

Name	Date modified	Type
1. Introduction	24-06-2020 17:06	File folder
2. Recording the Jmeter Scripts	24-06-2020 12:34	File folder
3. How to put load and analyse performance metrics	24-06-2020 12:40	File folder
4. Advanced Thread Group Methods for Real time load with Jmeter	24-06-2020 12:40	File folder
5. Http Cookie Manager to capture sessions	24-06-2020 12:40	File folder
6. Assertions in Jmeter	24-06-2020 12:41	File folder
7. Sample Application to Practice Jmeter Tool	24-06-2020 12:41	File folder
8. Jmeter controllers for Modules wise Metrics	24-06-2020 12:41	File folder
9. Timers in Jmeter	24-06-2020 14:22	File folder
10. Importance of Regular expressions in Jmeter	24-06-2020 12:12	File folder
11. Data Driven testing with Jmeter	24-06-2020 12:13	File folder
12. Handling Dynamic Responses -Advanced Jmeter-2	24-06-2020 12:15	File folder
13. Jmeter validations in Non GUI Mode	24-06-2020 12:18	File folder
14. Jmeter Distributed Mode in running tests in Slave machines	24-06-2020 12:19	File folder
15. Beanshell Scripting basics for Jmeter	24-06-2020 12:21	File folder
16. Integration of Selenium with Jmeter	24-06-2020 12:23	File folder
17. REST API Load Testing using Jmeter	24-06-2020 12:26	File folder
18. Monitoring Server performance	24-06-2020 12:27	File folder
19. Bonus lecture Bugzilla - Testing tool for Defect Tracking	24-06-2020 12:31	File folder
20. Jmeter older version topics(Obsolete)	24-06-2020 12:40	File folder
[FreeCourseLab.com]	24-06-2020 14:22	Internet Shortcut

**For any help:**

- [info@qaclickacademy.com](mailto:info@qaclickacademy.com)
- <https://www.udemy.com/user/testing-minds/>

**For Practice:**

<http://localhost:1080/webtours/home.html>

<http://newtours.demoaut.com/>

<https://blazedemo.com/>

<https://the-internet.herokuapp.com/login>

---

---

**=====Study: Day-1 (June 24, 2020)=====**

---

---

**=====1. Introduction=====****JMeter -**

- Load (1 type of Performance) Testing Tool

**Types of Performance testing -**

- Load
  - Stress
  - Volume
  - Endurance
- 

**Load testing** is the simplest form of Performance testing.

**Why is Performance testing needed?**

- To ensure the software applications will perform well under the particular workload.
- Example - Independence day sale: 1000 users are trying to order the same product at a same time (Amazon server will get hit as per our need)
- You need to win the Game (need to handle the traffic; load has to be handled)

Scripts have to be written here as well, but the flow will be working behind the scene (Headless mode). You won't have any UI open.

In the case of Selenium, the script runs for a single User. But, here, with JMeter, we can configure the script for n no. of Users (as per our requirement).

There are so many tools in the market used for doing performance testing.

**JMeter, LoadRunner, Neo Load, Load UI** and so many.

---

**Why JMeter?**

- Open source
- Cross platform support (Built on Java)
- GUI is friendly
- Scripting is not essential to learn (Manual tester can also go for it; no need to have any programming language knowledge)
- Lot of websites available

## **Download Apache JMeter**

- [http://jmeter.apache.org/download\\_jmeter.cgi](http://jmeter.apache.org/download_jmeter.cgi)
- Download the Binaries
- [apache-jmeter-5.3.zip](#)(WIN), [Apache-jmeter-5.3.tgz](#) (Linux)

## **Run JMeter**

- Navigate to path - D:\Work in Local machine\Softwares\apache-jmeter-5.3\bin
- ApacheJMeter.jar - Open this

---

**Test Plan** - (Actions + Components) (need to execute your performance test script)

## **WorkBench** (Removed from current version) -

- Practice area/Temporary storage (Rough work area)
- Components of WorkBench are not saved along with the Test plan

---

In JMeter, we can not have n no. of projects together (as Projects are there in Eclipse)  
Can work on 1 project at a time.

## =====2. Recording the Jmeter Scripts=====

### **Record** -

- Write a test
  - i) Go and perform the actions on Browser
  - ii) JMeter will listen all the actions and generate the test for us

---

### **Steps-**

- Right click on Test plan
- Add -> Non-Test elements -> HTTP(S) Test Script Recorder

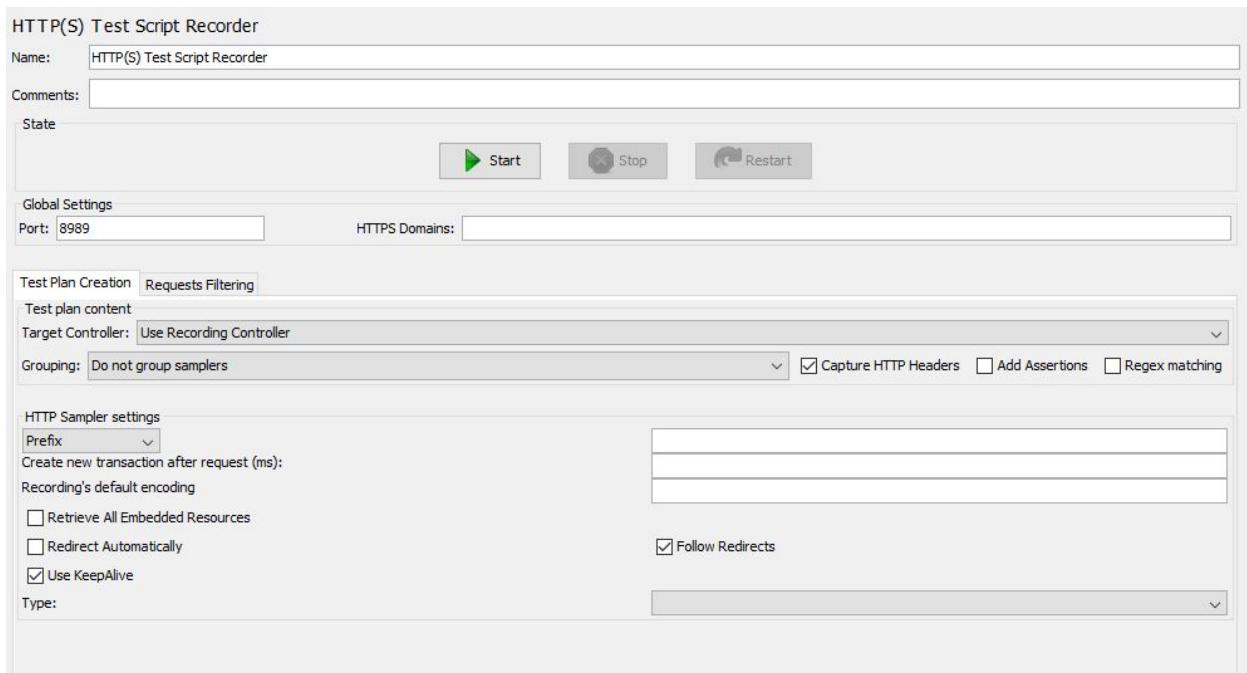
---

NOTE: Non-Test elements mean these are not the part of our test.

---

## **HTTP(S) Test Script Recorder -**

- Port - 8888 (default)
- HTTPS Domain - Optional field
- Test plan creation -
  - Target controller: -> Test Plan -> HTTP(S) Test Script Recorder
- Request Filtering -
  - URL Patterns to Exclude -> Add suggested excludes (All these types of files are not necessary for Performance Testing)

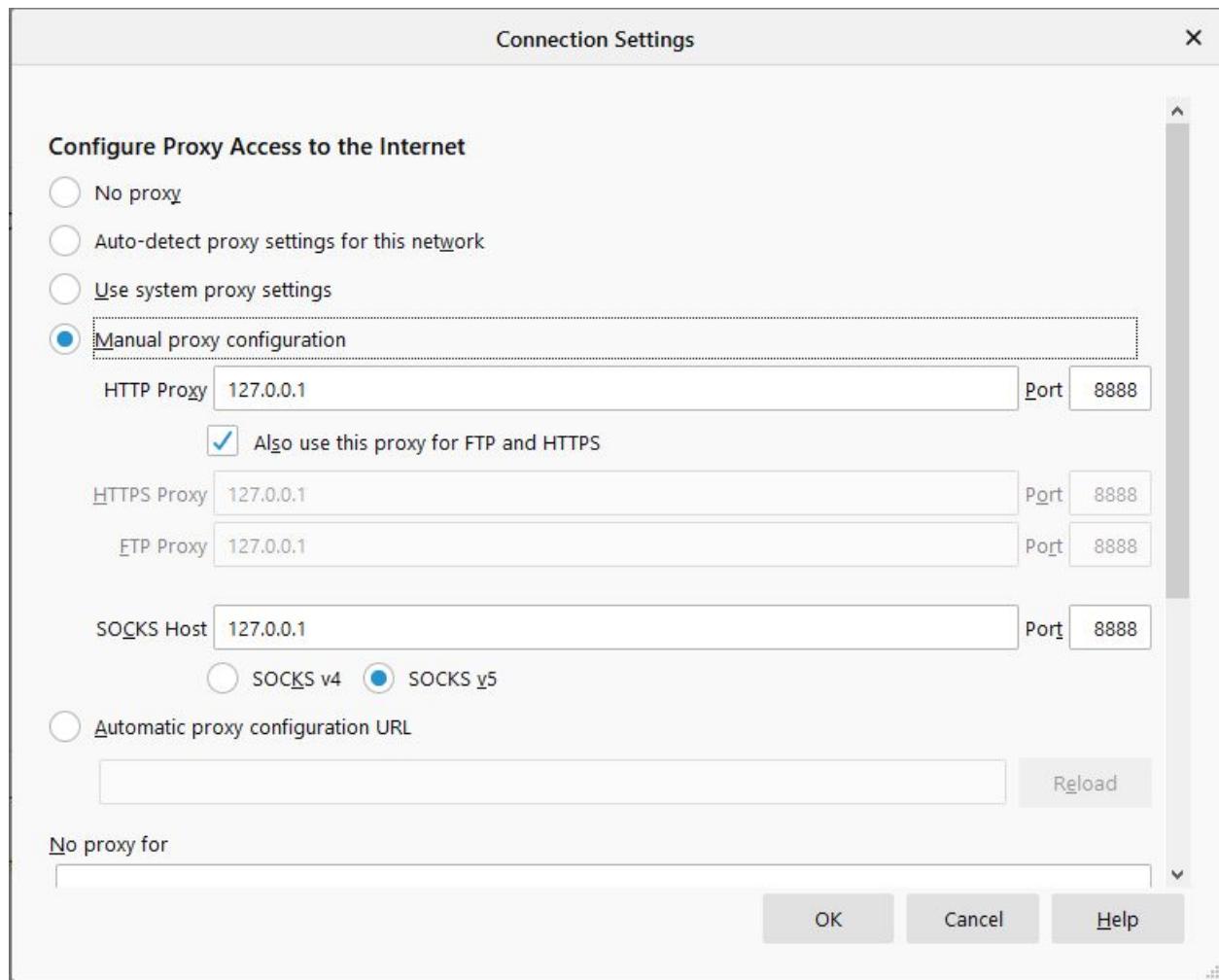


This setting is about Tool-side.

Now, I need to do setting in Browser side.

## FireFox -

- Options -> Network Proxy -> Settings ->
- Manual proxy configuration -> Port (8888) (This port number has to be same as in JMeter)
- HTTP Proxy: 127.0.0.1, and Port: 8888
- Enable the checkbox “**Use this proxy server for all protocols**”
- Now, hit the URL (<https://www.google.com>) -> The proxy server is refusing connections
- Browser will start working when you will start recording from JMeter



- For Record and Play, the above configuration has to be done
- For Recording (with https), Additional certificate (provided by JMeter) has to be installed in the Browser.  
Only, http is fine.
- Options -> Search for Certificates-> **View Certificates** -> Authorities -> import ->
- D:\Work in Local machine\Softwares\apache-jmeter-5.3\bin ->  
**ApacheJMeterTemporaryRootCA** -> Enable all the checkboxes

 certifi ×

## Search Results

### Certificates

When a server requests your personal certificate

- Select one automatically  
 Ask you every time

- Query OCSP responder servers to confirm the current validity of certificates

certifi

[View Certificates...](#)[Security Devices...](#)

## Search Results

### Certificate Manager

x

#### Certificates

When a server requests your personal certificate

- Select one automatically  
 Ask you every time

- Query OCSP responder servers to confirm the current validity of certificates

Your Certificates

People

Servers

Authorities

You have certificates on file that identify these certificate authorities

Certificate Name	Security Device
✓ _ JMeter Root CA for recording (INST...)	
_ JMeter Root CA for recording (I... Software Security Device	
▼ AC Camerfirma S.A.	
Chambers of Commerce Root - 20... Builtin Object Token	
Global Chambersign Root - 2008 Builtin Object Token	
▼ AC Camerfirma SA CIF A82743287	

[View...](#)[Edit Trust...](#)[Import...](#)[Export...](#)[Delete or Distrust...](#)

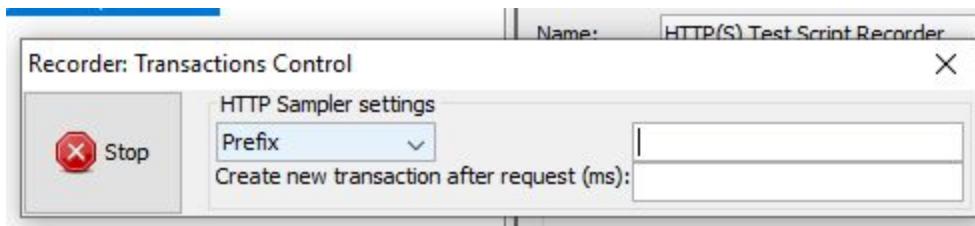
OK

**NOTE:** This certificate is imported to deal with websites using https.

And, if this pop-up comes even after clicking on Start, then, don't worry, keep it aside.

It won't interrupt recording.

And, once you are done with recording, change the proxy settings back to normal.



#### Steps -

- FireFox -> <https://www.blazedemo.com>
- Choose Origin and Destination and click on any Flight searched.
- Then, Stop recording.
- JMeter UI -> You can check each and every action recorded under HTTP(S) Test Script Recorder
- No. of points recorded are not even of use; they were recorded because of the Browser (FireFox) plugin. You can remove those extra steps as well.

**NOTE:** You won't see any UI in JMeter, it's all about Request and Responses.

---

### **Chrome Browser -**

- Add Chrome extension (**BlazeMeter | The Continuous Testing Platform**)
- Click on Extension added ->
- You need to have an account to start using this plugin (Sign in by any mode)
- Blazemeter benefit: It won't create no. of unnecessary files (only real and actual actions are recorded)

### **Start recording -**

- Forget about JMeter and Proxy settings for now and play with this extension
- Create first test using this extension
- This extension will record all the actions and generate a script at the end
- That script generated, you can import into JMeter (JMeter accepts file having extension as **.jmx**)
- Click on Extension added ->
- Enter the name of test: DEMO
- Click on Red button to start recording
- <https://www.blazedemo.com>
- Choose Origin and Destination and click on any Flight searched.
- Then, Stop recording.
- Save that file for JMeter (Extension- **.jmx**)

### **Import Script in Jmeter -**

- Save that earlier work first (File -> Save -> FireFox\_Demo)
- File -> Open -> Downloads (Location of file downloaded)
- Under ThreadGroup, you'll get all the actions recorded

Out of 2 ways to record, Blazemeter is easy and more convenient to use.

JMeter is used:

- To put 1000 load on the User flow (write your own script or just record)

---

---

**=====Study: Day-2 (June 25, 2020)=====**

---

---

**=====3. How to put load and analyse performance metrics=====**

**Thread Group -**

- Set of threads/Users executing the same scenario
- N no. of USERS to go with same flow
- Plays major role in JMeter
- N no. of combinations are available

**Steps -**

- We will put load (using Thread Group) this time on the same BLAZEDEMO flow.
- New window of JMeter -> Right click on Test Plan
- Add -> Non-Test elements -> HTTP(s) Test Script Recorder
- Do complete booking on Blazedemo
- Then, Stop recording

Now, we are ready with 1 of the test plans created with us.

We will add the Thread Group concept now.

- Right click on Test plan -> Add ->
  - Threads(Users) -> Thread Group
-

## **Configuration for Thread Group -**

- **No. of Threads (Users) : 50**
- **Ramp-up period (in seconds) : 5**
  - a. 1st second -> 10 users will hit the server,
  - b. 2nd second -> another 10 users will hit the server and so on.
- **Loop count : 2**
  - a. 2 times, this whole flow will go on
  - b. All these 50 Users will go through the flow for 2 times
- **Infinite : ENABLE**
  - a. Specify Thread Lifetime -
    - i. **Duration (seconds) : 60**
      - 1. If All 50 Users have completed the flow in 10 seconds, then, the loop will continue until the execution reaches 60 seconds
- **Action to be taken after a Sampler Error -**
  - This is used in case of any failure
  - What if the 4th User gets failed, then, what to do?

## **Steps -**

- You need to copy all the actions from HTTP(s) Test Script Recorder and paste it under Thread Group
- Save and then, Run the test

## **How to change the name of request?**

- Click on the request
- In the centre window, you'll get an option to rename it.

---

But, Where are the reports?

How will you analyze the status/outputs?

To do that, we need to use **Listeners**.

### **Listeners -**

- Listens to the execution
- 3 majorly used Listeners are
  - View Results Tree
  - Aggregate Report
  - Graph results

### **Steps -**

- Right click on Thread Group
- Add -> Listener
- View Results Tree, Aggregate Report, Graph results
- Run the test
- Reports are generated

In JMeter, you will not see any UI.

Behind the scenes, only APIs will be called.

---

### **Reading the reports -**

#### **1) View Results Tree -**

a)  -> **PASS**

Doesn't mean that test is always passed

Why?

Possibilities:

- Status code: 200, Response different (may be the wrong one)  
(Sometimes, need to check Request and Response data as well)
  - Add Employee
    - Employee added successfully - Status code (200)
    - Employee already exists - Status code (200)

b) Red marks -> **FAIL**

Request and Response data; you can check for each and every request.

The screenshot shows the JMeter interface with the 'Sampler result' tab selected. On the left, a tree view under the 'Text' category lists various requests, with '/reserve.php' highlighted in blue. The main panel displays detailed results for this request:

```

Thread Name: Thread Group 1-31
Sample Start: 2020-06-25 17:46:41 IST
Load time: 449
Connect Time: 0
Latency: 442
Size in bytes: 1527
Sent bytes: 514
Headers size in bytes: 256
Body size in bytes: 1271
Sample Count: 1
Error Count: 0
Data type ("text"|"bin"|""): text
Response code: 200
Response message: OK

HTTPSampleResult fields:
ContentType: text/html; charset=UTF-8
DataEncoding: UTF-8

```

## 2) Aggregate Report -

- If everything gets passed, then only, come to Aggregate Report to get information about the Parameters.  
Because, even if it fails, the Aggregate Report will have some parameters with it.
- Much about Parameters

The screenshot shows the JMeter interface with the 'Aggregate Report' tab selected. On the left, there are links for 'View Results Tree', 'Aggregate Report', and 'Graph Results'. The main panel displays a table of performance metrics:

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %
/reserve.ph...	1443	690	517	894	1050	3550	436	15842	
/purchase.p...	1416	704	520	735	945	4578	441	21709	
/confirmatio...	1371	661	522	699	872	3895	437	14840	
/reserve.php	740	666	550	870	987	2807	436	6897	
/purchase.php	722	607	549	766	857	1178	448	5110	
/confirmatio...	708	664	556	768	889	4467	448	6684	
TOTAL	6400	672	530	772	964	3800	436	21709	

## Home Work:

- Do a recording and check reports without putting any load on the Script.

## Aggregate report -

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min
/reserve.ph...	1443	690	517	894	1050	3550	436
/purchase.p...	1416	704	520	735	945	4578	441
/confirmatio...	1371	661	522	699	872	3895	437
/reserve.php	1362	734	529	931	1465	5255	405
/purchase.php	1325	680	528	817	1170	4852	404
/confirmatio...	1286	702	531	808	1116	5351	405
TOTAL	8203	695	523	813	1121	4880	404

- **Samples** - no. of users hit a particular service (Reserve -> 1443)
  - **Average** (measured in milliseconds)- average time taken by all the samples (to hit the request and to get the response) to execute specific label
    - Example: 3 users
      - 1st user - 15 milliseconds
      - 2nd user - 23 milliseconds
      - 3rd user - 18 milliseconds
      - Average =  $15+23+18/3 = 18.6$  milliseconds
  - **Min** - Minimum time taken by User to get a response
    - Example:
      - 1st user - 15 milliseconds
      - 2nd user - 23 milliseconds
      - 3rd user - 18 milliseconds
      - Min. time = 15 milliseconds
  - **Maximum** - Maximum time taken by User to get a response (Worst case situation)
    - Example:
      - 1st user - 15 milliseconds
      - 2nd user - 23 milliseconds
      - 3rd user - 18 milliseconds
      - Max. time = 23 milliseconds
  - **Error %** - % of failed requests per label (3-4 % is ok with 10000 users)
-

=====

**Study: Day-3 (June 26, 2020)**

=====

- **Throughput -**

- It is nothing but the capability of the Server.
- Very important to calculate Performance of the application
- No. of requests that are processed per time unit (sec, min, hours) by the server.
- This time is calculated from the start of the sample to the end of the last sample.
- Larger Throughput is good/better
- Samples - 1443, Throughput - 17.0/sec : Means when 1443 requests hitting are hitting reserve API, server is able to handle 17.0 requests per sec.
- **Server's CAPABILITY - 17.0**

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput
/reserve.ph...	1443	690	517	894	1050	3550	436	15842	0.00%	17.0/sec
/purchase.p...	1416	704	520	735	945	4578	441	21709	0.00%	16.7/sec
/confirmatio...	1371	661	522	699	872	3895	437	14840	0.00%	16.5/sec
/reserve.php	2188	696	519	886	1401	4556	402	7661	0.00%	1.6/min
/purchase.php	2129	641	523	773	962	3699	404	7291	0.00%	1.6/min
/confirmatio...	2073	646	519	763	942	4722	405	7935	0.05%	1.5/min
TOTAL	10620	671	520	786	1022	4410	402	21709	0.01%	7.8/min

- Change the Load (make it 100 from 50) and clear the previous results in Aggregate report
- Then, run it again and check the output in Aggregate report.

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput
/reserve.php	1269	758	503	913	1714	6622	400	10927	0.00%	36.7/sec
/purchase.php	1236	802	505	795	1892	7476	410	10287	0.00%	36.8/sec
/confirmatio...	1203	753	503	764	2096	7075	407	9741	0.00%	36.4/sec
TOTAL	3708	771	503	851	1923	7162	400	10927	0.00%	107.0/sec

- Now, you see, Throughput is increased even when no. of Users were increased.
- **What does that mean?**
  - That means Server is able to handle the more requests.
  - Performance of the application is Good.
  - **Another Possibility -**
    - Test plan is executing and Throughput is on constant value, that means, the server has come up on its max stage.

- That is the highest performance of Server to respond to the requests
  - **How to report to Client?**
    - With 100 Users of load, Throughput is 36.7 req/sec for reserve.php
- 

- **90% line -**

- 90% of the samples took no more than this time.
- 90% of 3258 Users = 2932.2 = Approx. 2932 Users
- **2932 Users** didn't take more than 2659 milliseconds to get the response

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput
/reserve.php	3258	1334	543	2659	6051	13561	400	19370	0.00%	6.0/sec
/purchase.php	3121	1277	533	2824	5751	13154	408	19696	0.00%	5.7/sec
/confirmatio...	2989	1229	524	2556	5089	13791	405	19519	0.00%	5.5/sec
TOTAL	9368	1282	532	2684	5751	13561	400	19696	0.00%	17.2/sec

- **Median -**

- It is the time in the middle of a set of samples result
  - It indicates that 50% of the samples took no more than this time, i.e, the remainder took at least as Long.
-

## Graph results -

- Important factor is **Deviation** here.
- **Deviation:** This shows the set of exceptional cases which were deviating from the average value of sample response time.
- When Deviation line is straight, it means our tests are not deviating from **Average sample response time**.
- Lesser the value of Deviation means the tests are more consistent (more stable application is)
- Deviation should be less than or equal to half of the average time for a label.



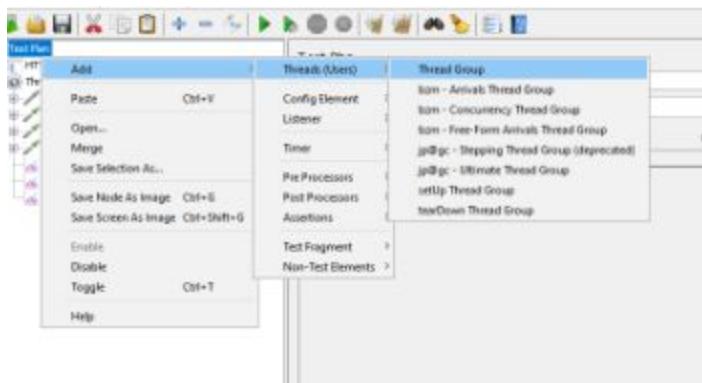
#### =====4. Advanced Thread Group Methods for Real time load with Jmeter=====

##### Additional Plugins for simulating Real Time Load -

- Navigate to <https://jmeter-plugins.org/wiki/PluginsManager/>
  - Download the Plugin manager (this will help us to download all the plugins from JMeter itself)
  - Download the JAR file and place it to below location :
  - D:\Work in Local machine\Softwares\apache-jmeter-5.3\lib\ext
  - Restart Jmeter -> Options -> Plugins manager
  - 1 more need to download -> **Custom Thread Groups** (will be available in Available plugins)
- 

Now,

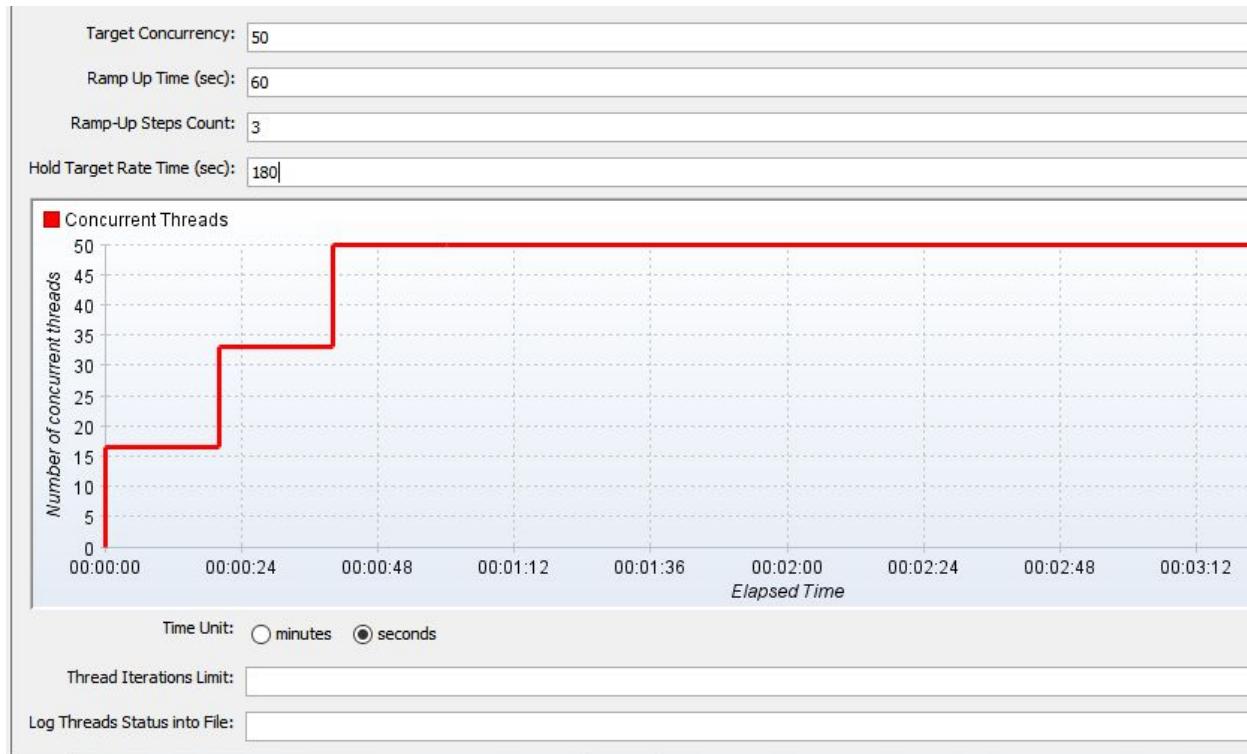
- Test Plan -> Add -> Threads (Users) -> more options available
- 2 important Thread Groups: **Concurrency** and **Ultimate**
- These Thread Groups and previously Thread group's functionalities are same, but, configurations are different



---

### Steps -

- Right click on Test plan -> Add -> Threads (Users) -> bzm - Concurrency Thread Group
- **Target Concurrency :** 50 (How many Threads you need to start with)
- **Ramp up time:** 60
- **Ramp-up Steps count:** 3 (Ramp those 50 users 3 times only)



---

---

---

**=====Study: Day-4 (June 29, 2020)=====**

---

---

### **Hold Target Rate time -**

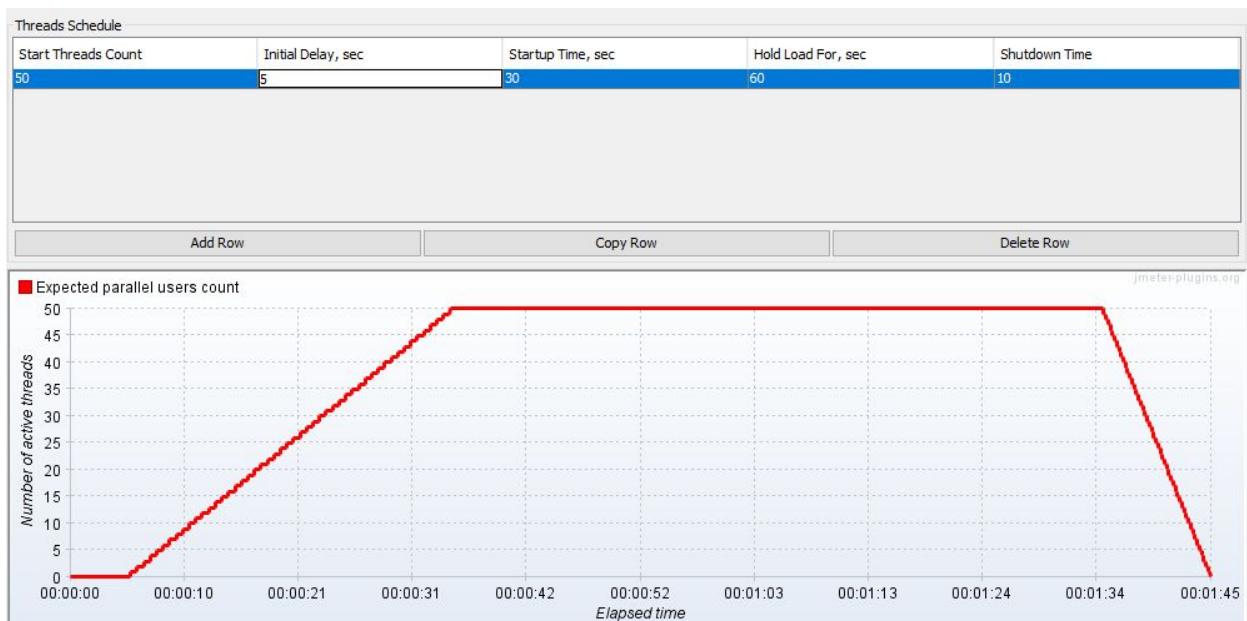
- How many seconds you want to hold this on server
- Straight line you can refer

### **Steps -**

- Copy all those steps in this newly created Concurrency Thread Group
- And, disable the previous Thread Group
- You can use the screenshots of this graph while reporting to the Client, what our Load Strategy is

### **Ultimate Thread Group -**

- Start Threads Count - 50
- Initial delay (sec) - 5
- Startup Time (Ramp-up time) - 30
- Hold Load For, sec
- Shutdown time(Ramp-down time) - 10 (Take 10 seconds to close all the 50 Users)



Additional Thread Groups have to be used to stimulate the Real-Time Users.  
It can vary from Project to Project.

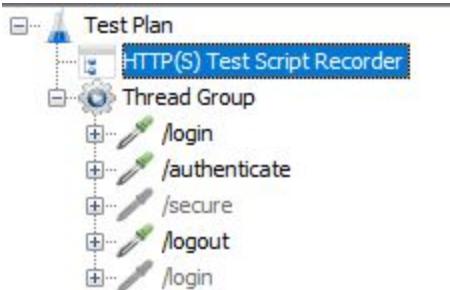
---

## =====5. Http Cookie Manager to capture sessions=====

### HTTP Cookie Manager -

#### Steps -

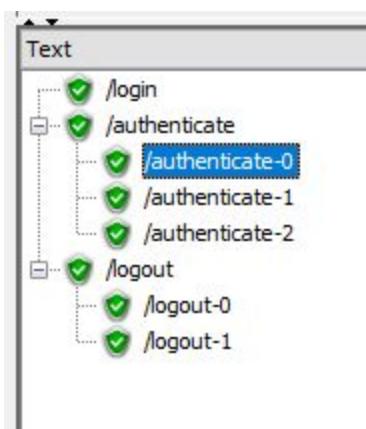
- Do start recording using JMeter and FireFox browser
- Navigate to <https://the-internet.herokuapp.com/login>
- Fill credentials
- Do Logout
- And, save it.
- Delete all the success.txt files from JMeter actions recorded (they are just Firefox related stuff)
- And, move all actions from Test script Recorder to Thread Group



- Add Listener (Results View Tree) and Run the Project.

Now, here is the Game.

Look like, script got passed successfully by observing Results View Tree.



## But is it really passed?

- Check /authenticate -> Response Data -> Response Body
- You'll get "**This is where you can log into the secure area. Enter.....**"

The screenshot shows a test results interface. On the left, there is a tree view of test cases under two main categories: '/authenticate' and '/logout'. Under '/authenticate', there are three entries: '/authenticate-0', '/authenticate-1', and '/authenticate-2', all marked with green checkmarks. Under '/logout', there are two entries: '/logout-0' and '/logout-1', also marked with green checkmarks. To the right of the tree view is a large window titled 'Response Body' which contains the following HTML code:

```
</div>
</div>
<div class="row">
<a href="https://github.com/touredave/the-internet"><img style="position: absolute; top: .png" alt="Fork me on GitHub"></a>
<div id="content" class="large-12 columns">
<div class="example">
<h2>Login Page</h2>
<h4 class="subheader">This is where you can log into the secure area. Enter <em>tomsmit</em> for the password. If the information is wrong you should see error messages.</h4>
<form name="login" id="login" action="/authenticate" method="post" >
<div class="row">
<div class="large-6 small-12 columns">
```

- But, on the UI, we are getting different stuff after gets successful login



## Secure Area

Welcome to the Secure Area. When you are done click logout below.

[Logout](#)

And, this is an **ISSUE**; The Playback has failed.

- It means, **don't just go with the iconic value of Results View Tree, you need to check the responses as well.**

Need to find out, what went wrong here, so, here, **CookieManager is going to figure out the stuff.**

Way to observe the overview of any request and response is **Sampler Result**.

1)

The screenshot shows the JMeter interface with the 'Sampler result' tab selected. On the left, a tree view under the 'Text' category shows requests for '/login', '/authenticate' (with three sub-requests: '/authenticate-0', '/authenticate-1', and '/authenticate-2'), and '/logout' (with two sub-requests: '/logout-0' and '/logout-1'). The '/authenticate-0' request is highlighted with a blue selection bar. The right panel displays detailed results for this request:

Sampler result	Request	Response data
Thread Name: Thread Group 1-1		
Sample Start: 2020-06-29 10:41:06 IST		
Load time: 283		
Connect Time: 0		
Latency: 283		
Size in bytes: 995		
Sent bytes: 577		
Headers size in bytes: 994		
Body size in bytes: 1		
Sample Count: 1		
Error Count: 0		
Data type ("text" "bin" ""): text		
Response code: 303		
Response message: See Other		

Below this, it says 'HTTPSampleResult fields:' followed by 'ContentType: text/html; charset=utf-8' and 'DataEncoding: utf-8'.

2)

The screenshot shows the JMeter interface with the 'Sampler result' tab selected. The tree view on the left is identical to the first one, with the '/authenticate-1' request highlighted. The right panel displays detailed results for this request:

Sampler result	Request	Response data
Thread Name: Thread Group 1-1		
Sample Start: 2020-06-29 10:41:06 IST		
Load time: 302		
Connect Time: 0		
Latency: 302		
Size in bytes: 966		
Sent bytes: 501		
Headers size in bytes: 965		
Body size in bytes: 1		
Sample Count: 1		
Error Count: 0		
Data type ("text" "bin" ""): text		
Response code: 302		
Response message: Found		

Below this, it says 'HTTPSampleResult fields:' followed by 'ContentType: text/html; charset=utf-8' and 'DataEncoding: utf-8'.

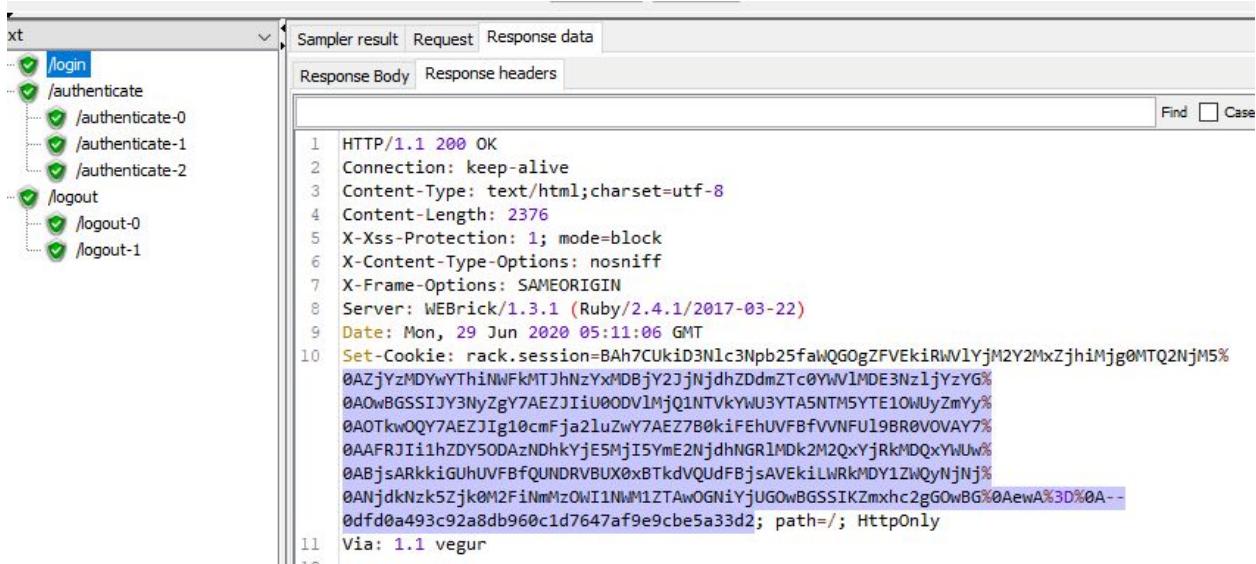
## Cookie -

- Example -
  - Hit facebook.com in Browser ->
  - HomePage (having Email and Password textboxes) will open
  - Facebook gives 1 Session cookie in the response (to identify the User uniquely)
  - This session cookie is saved by Browser (That's why when you open the facebook page, even after 3-4 days, you are logged in already. You don't need to provide credentials again and again)
  - Now, when you visit any page in the facebook (example - Profile, posts, Comments, Messages, etc.).....For every request, this session cookie is attached to requests
  - Cookie concept is mainly implemented in the secured applications

---

## What's happening in our case?

- 1) <https://the-internet.herokuapp.com/> - Gave 1 Session Cookie in the response
- 2) Session Cookie stored by browser
- 3) Give credentials -> https://the-internet.herokuapp.com/authenticate  
username=tomsmith&password=SuperSecretPassword  
[no cookies]
- 4) **NOTE:** Application is expecting some cookies, but, nothing is passed here by JMeter. (Browser passes the Cookies automatically when the request is hit)
- 5) **Reason:** JMeter does not store cookies like Browser.
- 6) But, you get Cookie in response here as well.

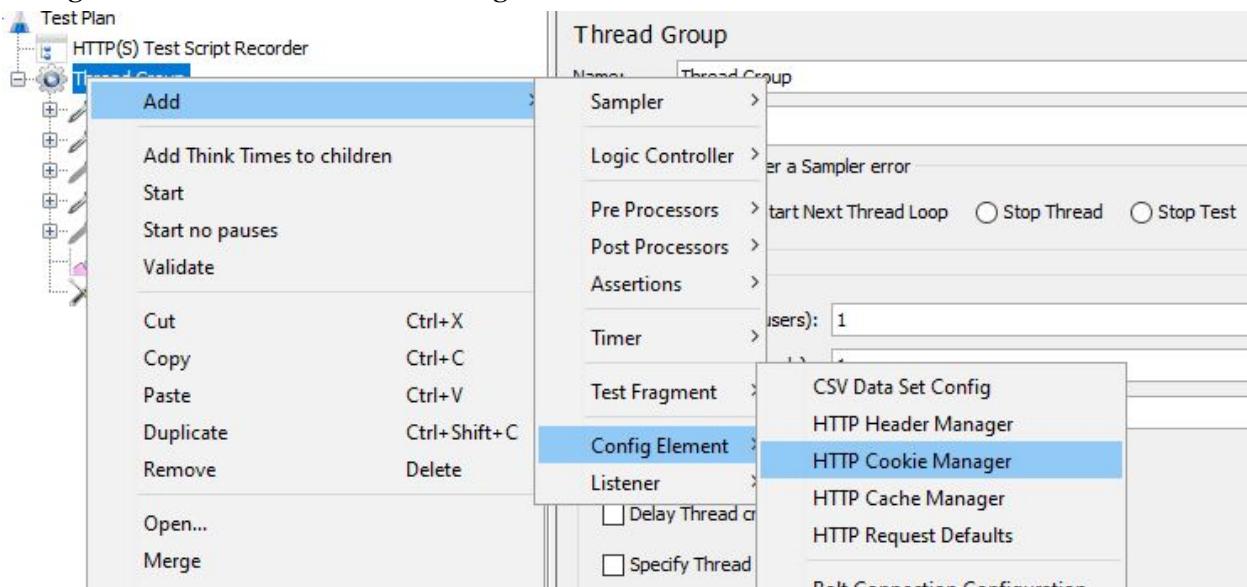


The screenshot shows the JMeter interface. On the left, there is a tree view of requests under a 'xt' test plan. The requests listed are: login, /authenticate, /authenticate-0, /authenticate-1, /authenticate-2, /logout, /logout-0, and /logout-1. On the right, the 'Response' tab is selected, showing the response body and headers. The 'Response Body' tab is active, displaying the HTTP response code and headers. The 'Response headers' tab is also visible. A specific line in the headers section, 'Set-Cookie: rack.session=... (long session ID)', is highlighted with a blue selection bar, indicating it is being examined.

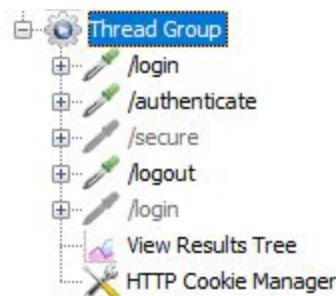
- 7) But, JMeter does not store it (because JMeter is not a Browser)
- 8) Explicitly, we need to do that. (Store it and then, attach it with the next request)
- 9) **HTTP Cookie Manager** (it is the one, we are going to use it)

### Steps for HTTP Cookie Manager -

- Right click on Thread Group -> Add
- Config element -> HTTP Cookie Manager



- Now, this Cookie manager will store and attach the Cookie(got in 1st API call) with all the further steps



- Now, save the project and run it; check for response now.

- Previous: no cookie is attached

Text

```

    ✓ /login
    ✓ /authenticate
      ✓ /authenticate-0
      ✓ /authenticate-1
      ✓ /authenticate-2
    ✓ /logout
    ✓ /login
    ✓ /authenticate
      ✓ /authenticate-0
      ✓ /authenticate-1
    ✓ /logout
  
```

Sampler result Request Response data

Request Body Request Headers

```

1 POST https://the-internet.herokuapp.com/authenticate
2
3 POST data:
4 username=tomsmith&password=SuperSecretPassword%21
5
6 [no cookies]
7
  
```

- Now: Cookie is attached

Text

```

    ✓ /login
    ✓ /authenticate
      ✓ /authenticate-0
      ✓ /authenticate-1
      ✓ /authenticate-2
    ✓ /logout
    ✓ /login
    ✓ /authenticate
      ✓ /authenticate-0
      ✓ /authenticate-1
    ✓ /logout
  
```

Sampler result Request Response data

Request Body Request Headers

```

1 POST https://the-internet.herokuapp.com/authenticate
2
3 POST data:
4 username=tomsmith&password=SuperSecretPassword%21
5
6 Cookie Data:
7 rack.session=BAh7CUkiD3Nlc3Npb25faWQG0gZFVEkiRTkyNTRjZTc0Y2JkNjEwMjgzYzQz%
  0AOGEExY2NjZDQ5NzNiYzEyMTM4NGE5NmQ5ZDI2YmI3NTEwMDJkMjU3YzhmMGIG%
  0AOwBGSSIJY3NyZgY7AEZZIiVmMD1lMWQyYjdjOTM4ZjY4ZmY5NDM3OWRkZWQw%
  0AOwISMQY7AEZZIg10cmFja2luZwY7AEZB0kfFehUVFBfVNNFU19BR0VOVAY7%
  0AAFRJII1hZDY5ODAzNdhkYjE5MjI5YmE2NjdhNGR1MDk2M2QxYjRkMDQxYWUw%
  0ABjsARKkiGUhUVFBfQUndRVBUX0xBTkdvQUdFBjsAVEkilLWRkMDY1ZWQyNjNj%
  0ANjdkNzk5Zjk0M2FiNmMzOWI1NwM1ZTAwOGNiYjUG0wBGSSIKZmxhc2gG0wBG%0AewA%3D%0A--
  09a7b1e8e8493d915520a15c8b65ddc8431fc698
8
  
```

And,

Text

```

    ✓ /authenticate-1
    ✓ /authenticate-2
  ✓ /logout
  ✓ /login
  ✓ /authenticate
    ✓ /authenticate-0
    ✓ /authenticate-1
  ✓ /logout
  
```

<div data-alert id='flash' class='flash success'>
 You logged into a secure area!
 <a href="#" class="close">&times;</a>
</div>

</div>
</div>
<div class="row">
 <a href="https://github.com/touredave/the-internet"><img style="position: absolute; top: 0; right: 0; border: none;" alt="Fork me on GitHub" /></a>
 <div id="content" class="large-12 columns">
 <div class="example">
 <h2><i class="icon-lock"></i> Secure Area</h2>
 <h4>Welcome to the Secure Area. When you are done click logout below.</h4>
 <a class="button secondary radius" href="/logout"><i class="icon-2x icon-signout"></i> Logout</a>
 </div>
 </div>
</div>
<div id="page-footer" class="row">
 <div class="large-4 large-centered columns">
 <hr>
 <div style="text-align: center;">Powered by <a target="\_blank" href="http://elementalselenium.com/">Ele
 </div>
</div>

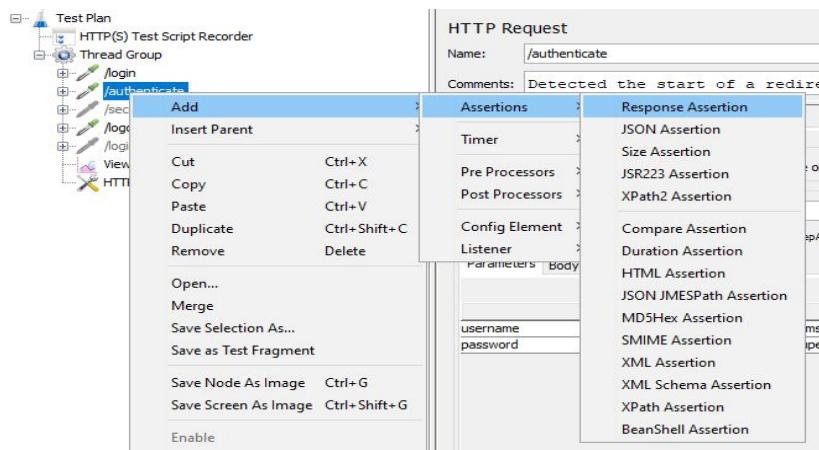
## =====6. Assertions in Jmeter=====

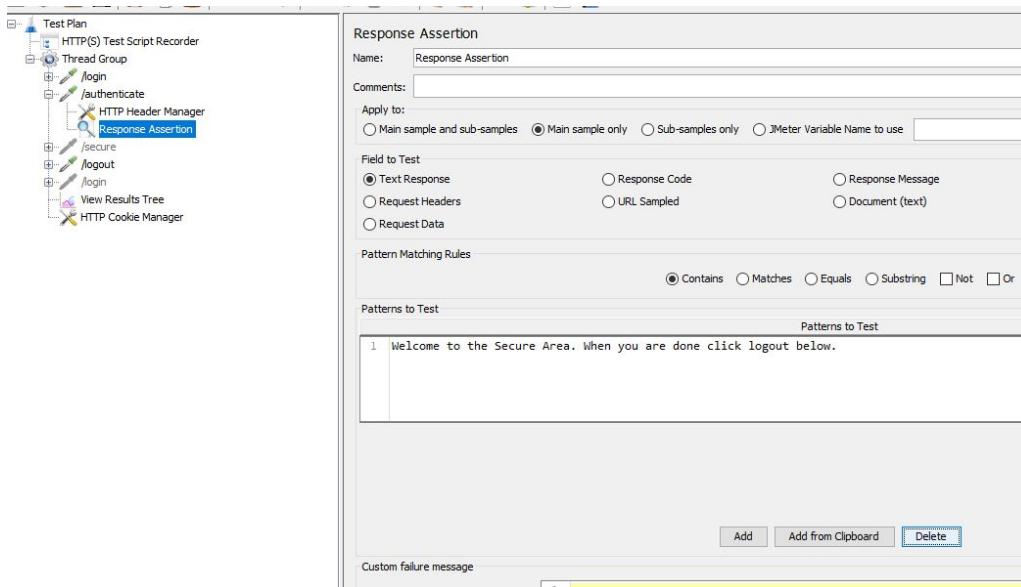
Till now, we were manually checking the Response Status codes and Response messages.

### Steps Assertion for /authenticate request -

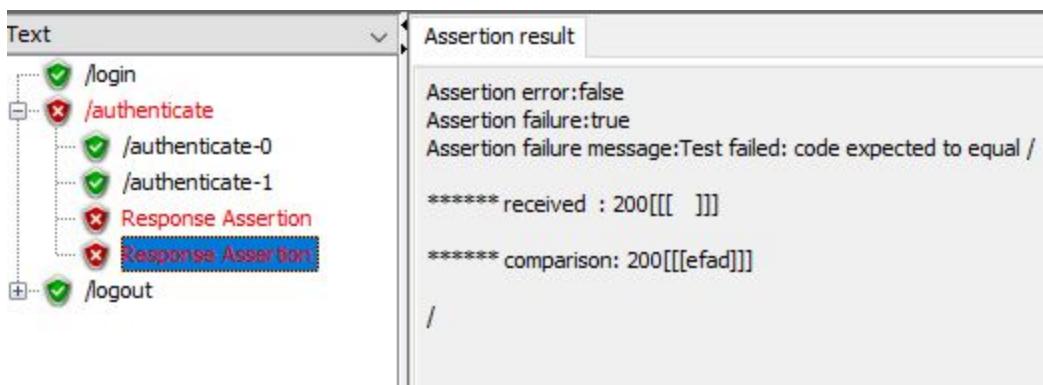
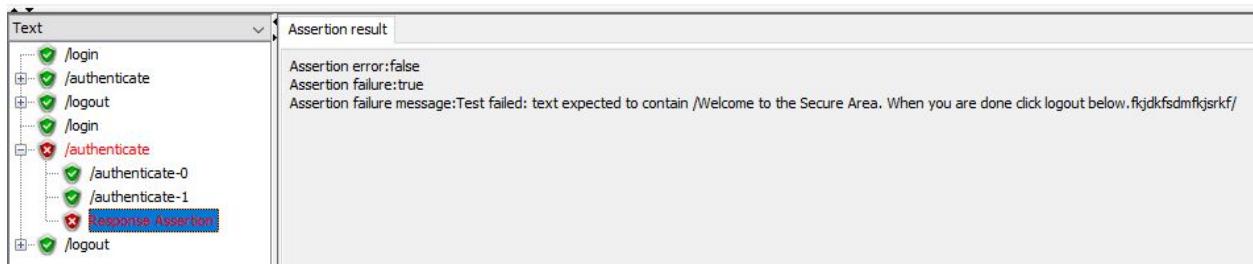
- Validation check: "Welcome to the Secure Area. When you are done click logout below"; has to be present in the Response Body
- Right click on /authenticate request -> Add -> Assertion
- Many types of Assertions are available (Mainly used for Web requests: Response Assertion, Size Assertion, Duration Assertion, HTML Assertion)

### Response Assertion -





You can add no. of Assertions to 1 request.



#### NOTE:

- 1 more listener is there which helps to understand the results of Assertions.
- **Assertion Results**

The screenshot shows the JMeter interface with the 'Assertion Results' tab selected. On the left, a tree view displays various assertion configurations: Response Assertion, /secure, /logout, HTTP Header Manager, Response Assertion, Response Assertion, /login, View Results Tree, HTTP Cookie Manager, and Assertion Results. To the right, a list of failed responses is shown:

```
/login
/authenticate
/logout
/login
/authenticate
/logout
```

Below this, another section shows a tree view with /login, /authenticate, and /logout, followed by an error message:

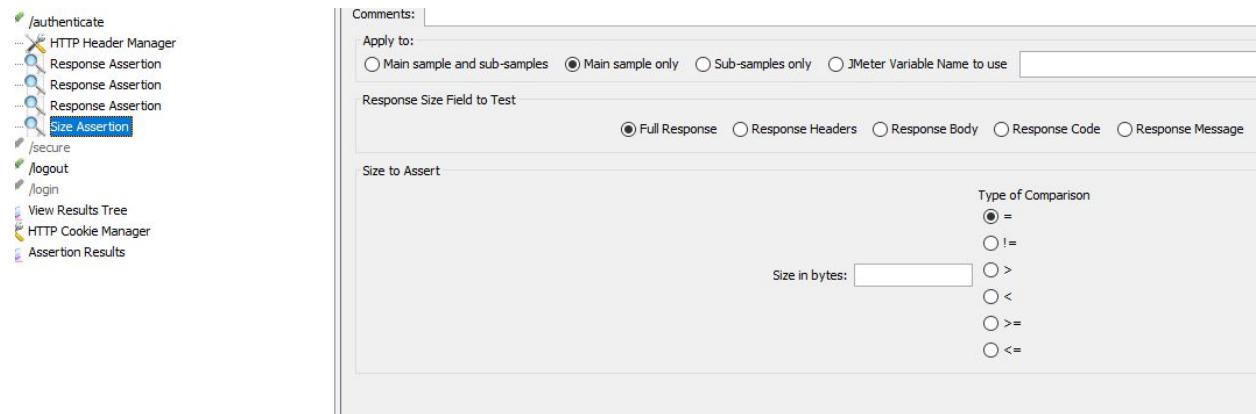
```
/login
/authenticate
/logout
Response Assertion : Test failed: headers expected to contain /Content-Type: text/html;charset=utf-8/
```

  
**/authenticate : Main Sample**  
**/authenticate-0 : Sub-Sample**  
**/authenticate-1 : Sub-Sample**

If you want to put assertions on sub-samples, then, it is also configurable.

The screenshot shows the 'Response Assertion' configuration dialog. It includes fields for 'Name' (set to 'Response Assertion'), 'Comments', and 'Apply to'. The 'Apply to' section contains four radio buttons: 'Main sample and sub-samples' (unchecked), 'Main sample only' (checked), 'Sub-samples only' (unchecked), and 'JMeter Variable Name to use' (unchecked).

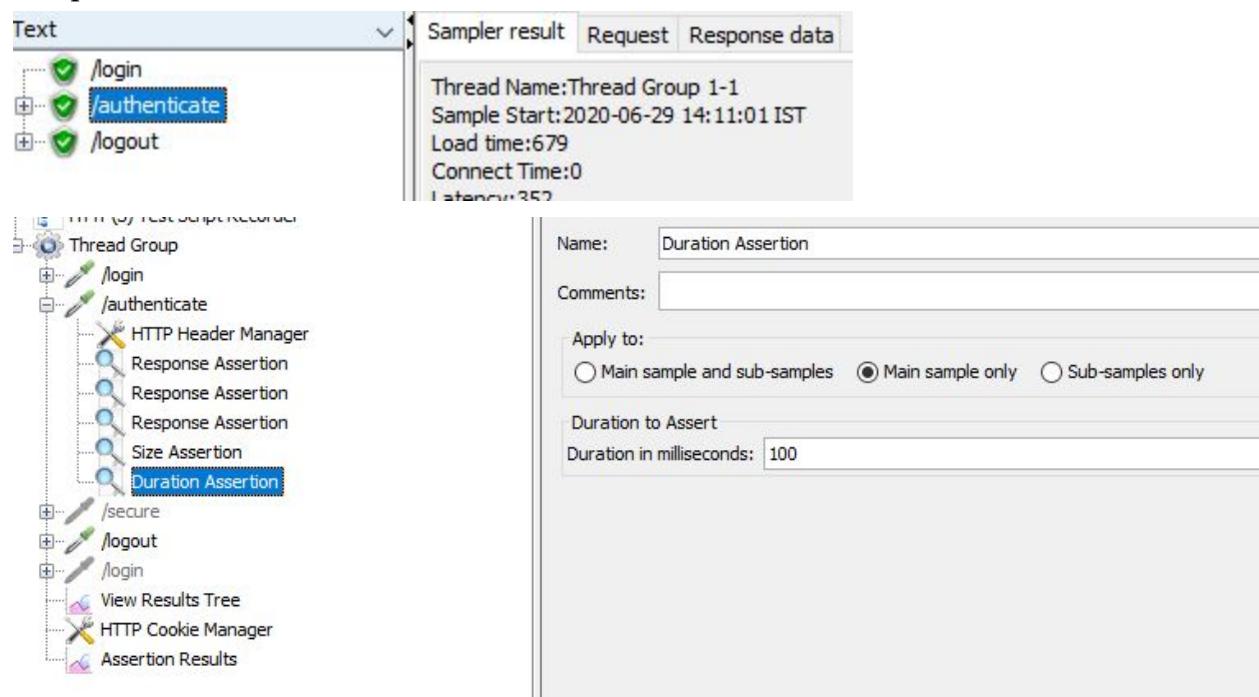
## Size Assertion -



- In Sampler results, you can find out the details of request size.

## Duration Assertion -

- How much time did authenticate service take to get the response from the Server?
- Sampler Results -> **Load time**



- Duration in milliseconds: 30 (means **maximum** the service should take 30 sec to respond)

## =====7. Sample Application to Practice Jmeter Tool=====

- HP has provided 1 application for doing practice on JMeter tool: **Web Tours application** (<https://marketplace.microfocus.com/appdelivery/content/web-tours-sample-application>)
- **Web Tours** is a sample Web-based travel agency application used to demonstrate how LoadRunner is used as a solution for performance testing.

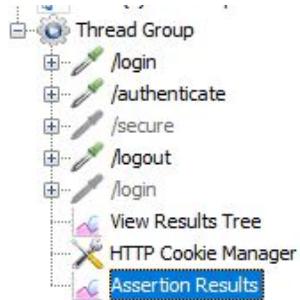
### Steps -

- Download this file - **Web Tours 1.0.zip** and extract this file
- Install this file
- D:\Work in Local machine\Softwares\Web Tours 1.0\WebTours\conf\httpd.conf
- D:\Work in Local machine\Softwares\Web Tours 1.0\WebTours\StartServer.bat
- Find : Listen -> Default Port: 1080 (you can change it if required)
- <http://localhost:1080/webTours/home.html>
- Do Signup with some details to create the User

## =====8. Jmeter controllers for Modules wise Metrics=====

### Controller -

- It is container/folder where all of your requests can be stored
- Example (Flight Reservation):
  - Login - **Login controller**
  - Scenario #1: Search for Flight and give required details and select the flight - 7 requests/samplers = 1 controller/container/folder - **Flight selection controller**
  - Scenario #2: Passenger details, Payment information and confirm booking - **Reservation controller**



/login, /authenticate, etc. all are requests/samplers.

- Each controller contains a set of request/samplers.

### Types of Controllers -

- 1) Transactional controller -
    - Clients may ask how much time the application is taking to do
      - Login
      - Reservation,
      - Payments
      - Confirmation code
    - Add -> Thread Group -> Add -> Logic controller -> Transaction controller
    - We can call this set of requests as "**Module**" as well.
-

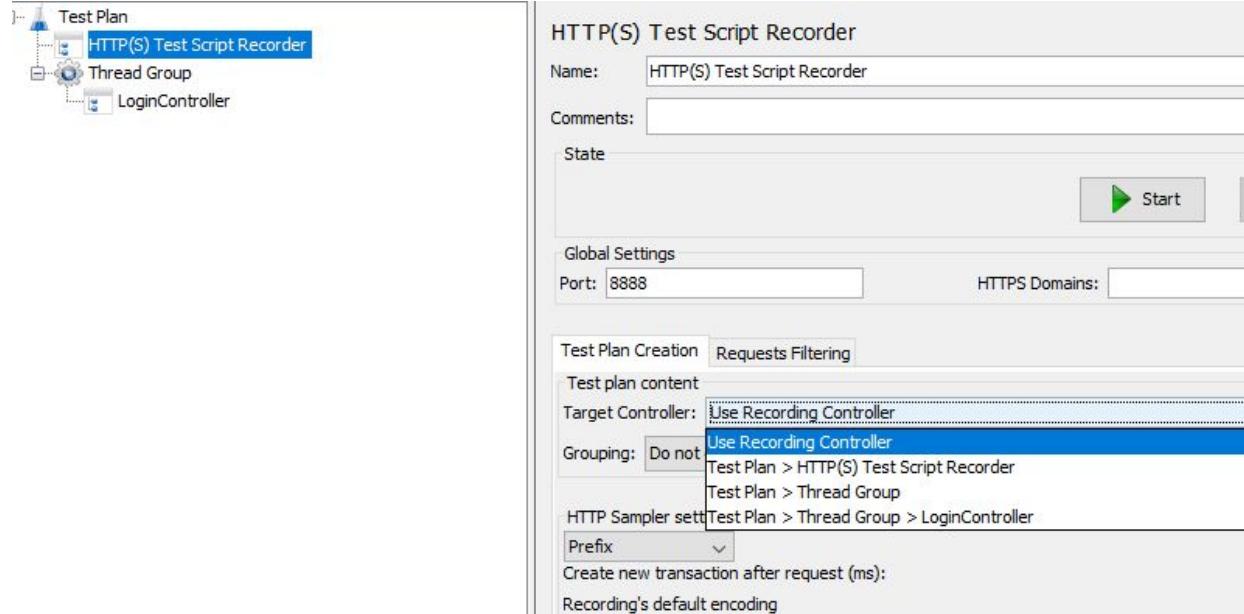
=====

**Study: Day-5 (June 30, 2020)**

=====

2) Recording controller -

- Specific controller used for recordings
- You can directly put all the steps/actions to be recorded inside specific recording controller instead of HTTP(s) Test recorder



**Steps -**

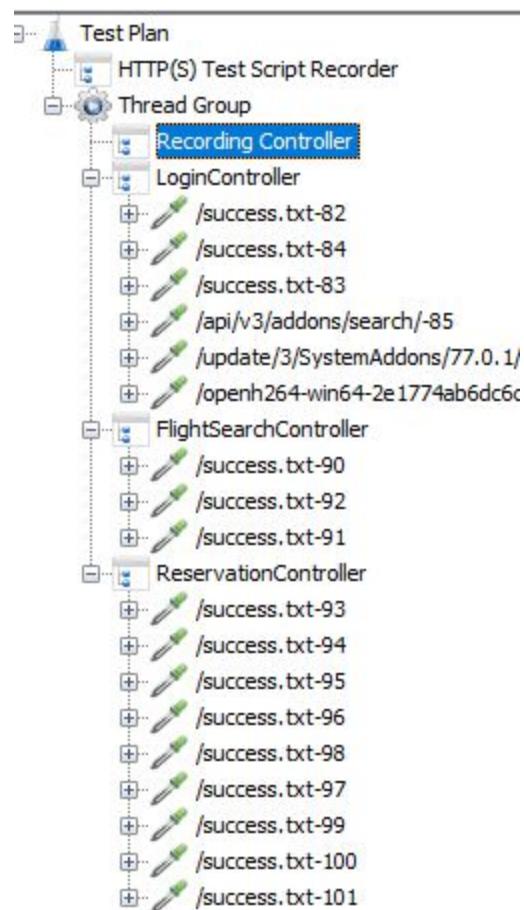
- Add -> Logic controller -> Recording Controller
- Do start recording in Recording Controller
- Firefox -> <http://localhost:1080/webTours/home.html> -> Do Login
- **NOTE:** If this above URL mentioned is not giving you the option to do Login, then, hit this URL: <http://localhost:1080/cgi-bin/welcome.pl?signOff=1>
- Create new Transactional Controller (LoginController) and move all above actions (related to Login) inside this.
- Right click on Thread Group -> Add -> Logic controller -> Transactional Controller
- Do record steps for Flights selection and put all those actions under new Transactional Controller (FlightSearchController)
- Complete the Booking and put all those actions under new Transactional Controller (ReservationController)

## Invoice

Thank you for booking through Web Tours.

RajatVerma's Flight Invoice		Tue Jun 30 15:14:03 2020
Flights Reservation Requests:		Cost
1. A Business Class ticket from Los Angeles to San Francisco. 07/01/2021 : 5pm : Flight 362 leaves Los Angeles for San Francisco.		\$82
* Total Cost		<b>\$82</b>
- Total Charged to Credit Card # 4444333322221111		<b>\$82</b>
* Credit Account Balance		<b>\$0.00</b>

[Book Another](#)



- We have created 3 controllers for Flight Booking

Now, let's add some listeners(View Results Tree, Aggregate Report) for observations.

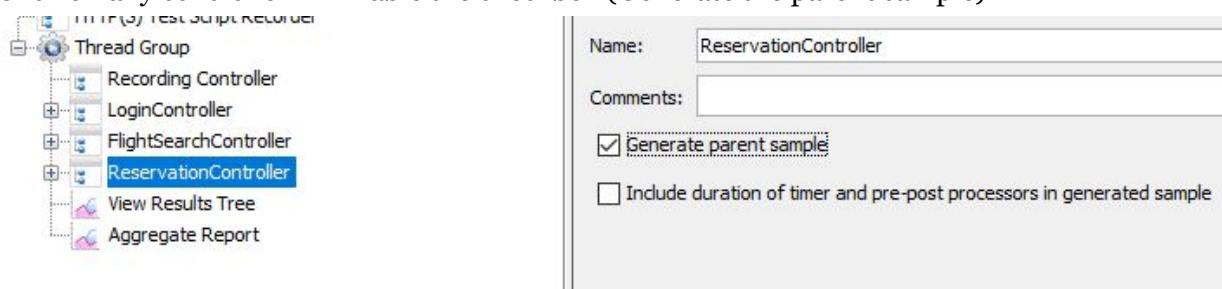
### Steps -

- Run it for only 1 User

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received K...	Sent KB/sec
/success.txt-82	1	440	440	440	440	440	440	440	0.00%	2.3/sec	0.85	0.6
/success.txt-84	1	65	65	65	65	65	65	65	0.00%	15.4/sec	5.77	4.5
/success.txt-83	1	85	85	85	85	85	85	85	0.00%	11.8/sec	4.41	3.4
/api/v3/add...	1	2141	2141	2141	2141	2141	2141	2141	0.00%	28.0/min	0.75	0.2
/update/3/S...	1	590	590	590	590	590	590	590	0.00%	1.7/sec	1.07	0.7
/openh264...	1	4777	4777	4777	4777	4777	4777	4777	0.00%	12.6/min	92.69	0.0
<b>LoginController</b>	<b>1</b>	<b>8098</b>	<b>0.00%</b>	<b>7.4/min</b>	<b>55.08</b>	<b>0.2</b>						
/success.txt-90	1	80	80	80	80	80	80	80	0.00%	12.5/sec	4.69	3.6
/success.txt-92	1	53	53	53	53	53	53	53	0.00%	18.9/sec	7.08	5.5
/success.txt-91	1	42	42	42	42	42	42	42	0.00%	23.8/sec	8.93	7.0
FlightSearch...	1	175	175	175	175	175	175	175	0.00%	5.7/sec	6.39	5.0

- In Aggregate report, you may see all the samplers/requests linked to Which Controller
- In the above case, 6 samplers are included inside Login controller; all these 6 samplers are treated as an 1 Transaction

- For each Controller, we can now check -
  - Average  
(Login: 8098 milliseconds)
  - Throughput  
(Flight search: 5.7/sec; means 5.7 requests are being served by the server in a sec)
- In case, if you don't all the samplers/requests details in the report, you may exclude that.
- **How?**
  - Click on any controller -> Enable the checkbox (Generate the parent sample)



- Now, run the project again and check the reports

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput
LoginController	1	4194	4194	4194	4194	4194	4194	4194	0.00%	14.3/min
FlightSearch...	1	190	190	190	190	190	190	190	0.00%	5.2/sec
Reservation...	1	882	882	882	882	882	882	882	0.00%	1.1/sec
TOTAL	3	1755	882	4194	4194	4194	190	4194	0.00%	34.1/min

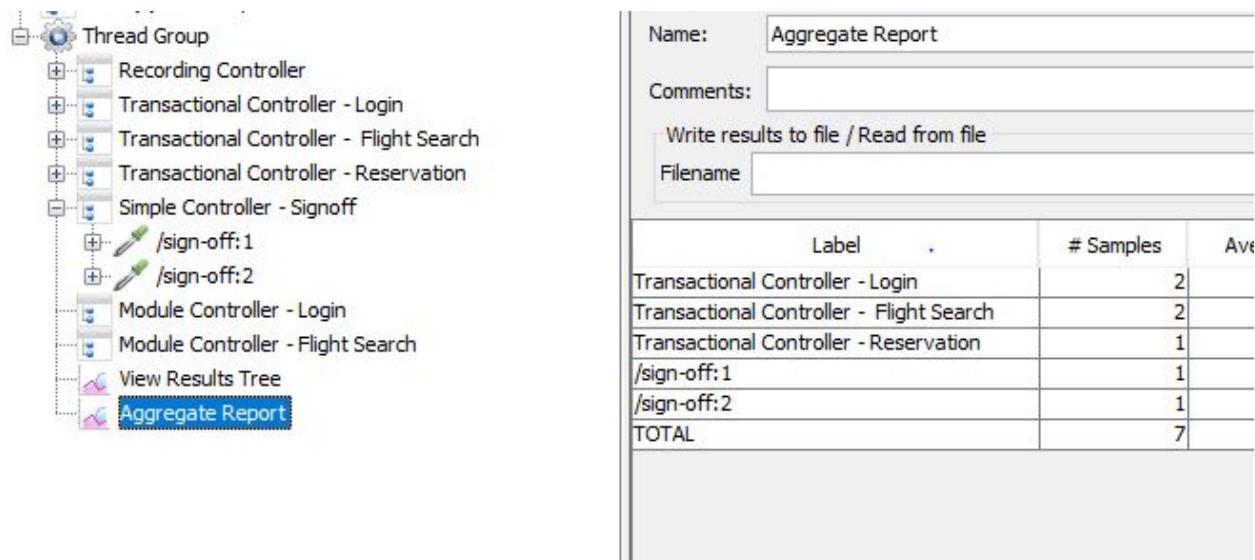
## Client -

- If asks How much time did Login functionality take to be finished with n no. of Users.
- Check the reports and, you are good to answer all your Client queries.

- He/She may ask you to divide some functionalities as 1 transaction
  - Example: Banking apps
    - Enter CC details and pay min. Amount - 1 transaction
    - Scheduling the payment for some specific date and select the funding source (from where the money has to be deducted) - 1 transaction
- 

### 3) Simple controller -

- Almost same as Transactional controller
- It is just a container; can contain n no. of samplers/requests, but can't keep track of any performance on transaction level
- As you are able to see controller level performance in aggregate report for Transactional controller, you won't be able to do that in Simple controller
- You just can put some actions/samplers/requests under that (As 1 folder is having no. of requests inside it)



The screenshot shows the JMeter interface with a test plan containing a Thread Group and various controllers. On the right, an 'Aggregate Report' dialog is open, showing performance data.

**Test Plan Structure:**

- Thread Group
  - Recording Controller
  - Transactional Controller - Login
  - Transactional Controller - Flight Search
  - Transactional Controller - Reservation
  - Simple Controller - Signoff
    - /sign-off:1
    - /sign-off:2
  - Module Controller - Login
  - Module Controller - Flight Search
  - View Results Tree
  - Aggregate Report

**Aggregate Report Dialog:**

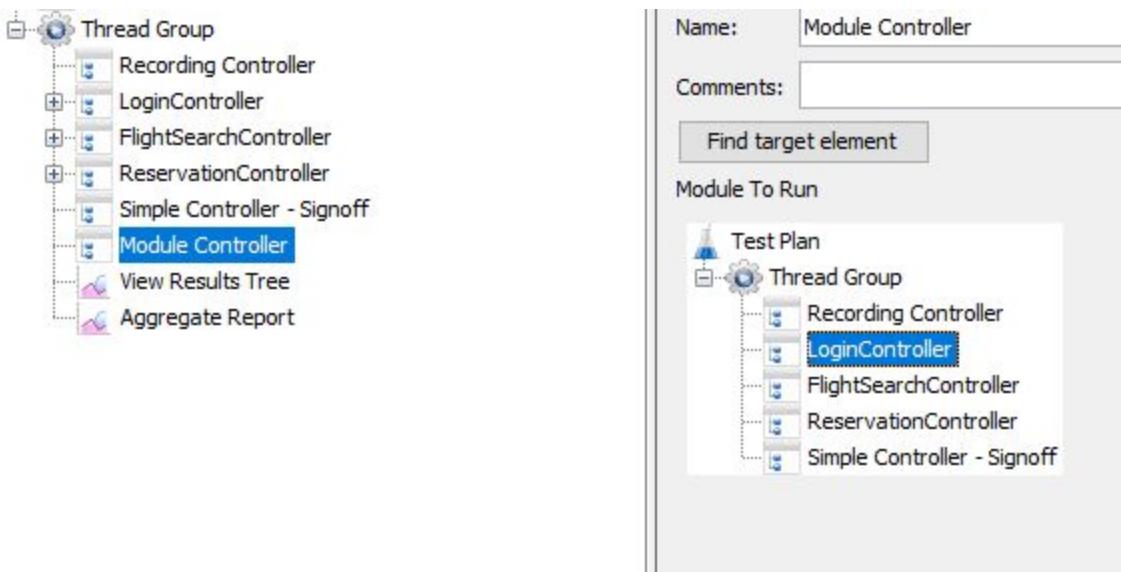
Name:	Aggregate Report																					
Comments:																						
Write results to file / Read from file																						
Filename:																						
<table border="1"> <thead> <tr> <th>Label</th> <th># Samples</th> <th>Avg</th> </tr> </thead> <tbody> <tr> <td>Transactional Controller - Login</td> <td>2</td> <td></td> </tr> <tr> <td>Transactional Controller - Flight Search</td> <td>2</td> <td></td> </tr> <tr> <td>Transactional Controller - Reservation</td> <td>1</td> <td></td> </tr> <tr> <td>/sign-off:1</td> <td>1</td> <td></td> </tr> <tr> <td>/sign-off:2</td> <td>1</td> <td></td> </tr> <tr> <td>TOTAL</td> <td>7</td> <td></td> </tr> </tbody> </table>		Label	# Samples	Avg	Transactional Controller - Login	2		Transactional Controller - Flight Search	2		Transactional Controller - Reservation	1		/sign-off:1	1		/sign-off:2	1		TOTAL	7	
Label	# Samples	Avg																				
Transactional Controller - Login	2																					
Transactional Controller - Flight Search	2																					
Transactional Controller - Reservation	1																					
/sign-off:1	1																					
/sign-off:2	1																					
TOTAL	7																					

#### 4) Module controller -

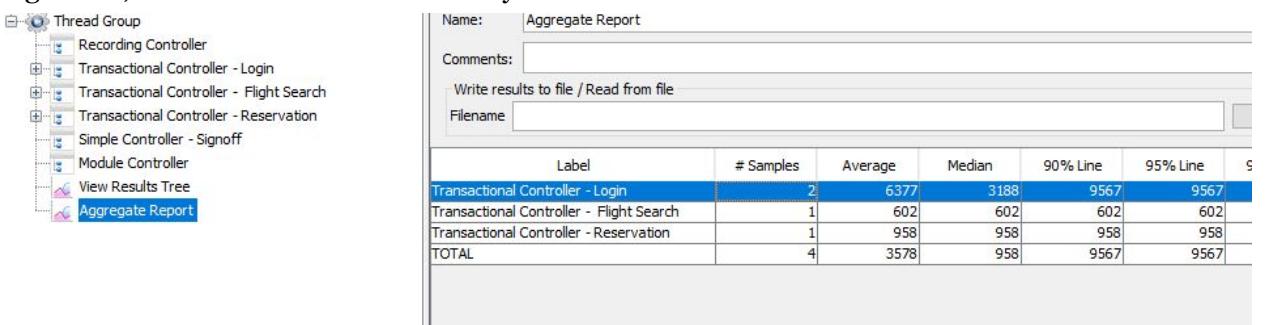
- If you want to repeat the Login functionality, you can just repeat those steps by using previous actions recorded.
- No need to record it again and again.
- Re-Usability: this has to be implemented

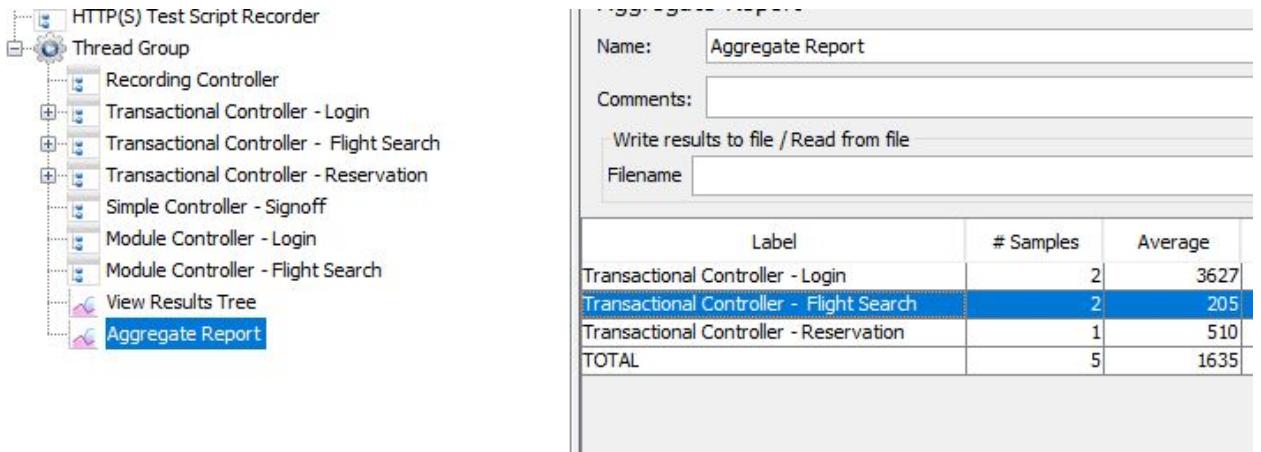
#### Steps -

- Right click on Thread Group -> Add -> Logic controller -> Module Controller
- Click on module controller -> Expand the Thread Group
- Select the controller (Which functionality you want to use again)



- In the Aggregate reports, it'll be shown as
- Login is called twice from our test plan, though Login functionality was called even after Sign - Off, but will be shown in row only





## 5) InterLeave Controller -

- Controller is nothing, but an container (contains set of requests/samplers)
- 1 sampler per iteration is executed (Top to Bottom order)
- Example: 3 samplers -> Only 1 out of 5 sampler will be picked
- Usage -
  - 1st iteration - 1st sampler
  - 2nd iteration - 2nd sampler
  - 3rd iteration - 3rd sampler

### Simple Interleave Example

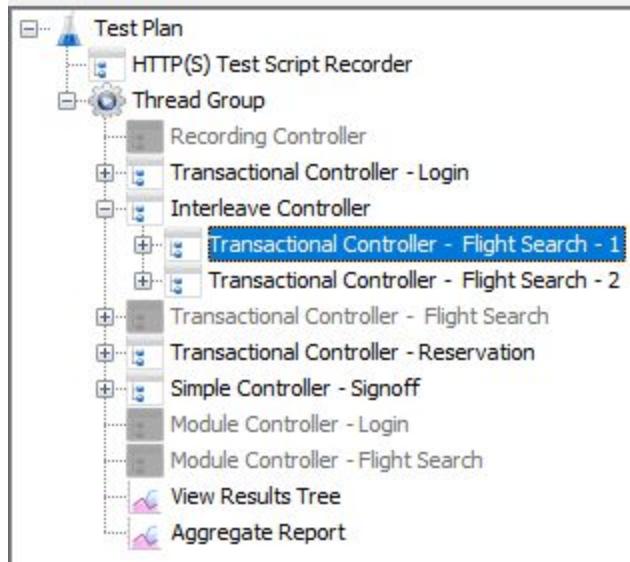
[Download](#) this example (see Figure 1). In this example, we configured the Thread Group to have two threads and a loop count of five, for a total of ten requests per thread. See the sequence JMeter sends the HTTP Requests.



Figure 1 - Interleave Controller Example 1

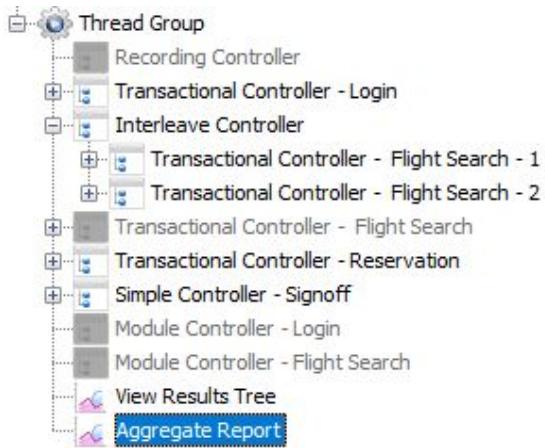
### Loop Iteration Each JMeter Thread Sends These HTTP Requests

1	News Page
1	Log Page
2	FAQ Page
2	<b>Log Page</b>
3	Gump Page
3	Log Page



- Flight Search 1 (Boston to Los Angeles)
- Flight Search 1 (New York to Dublin)

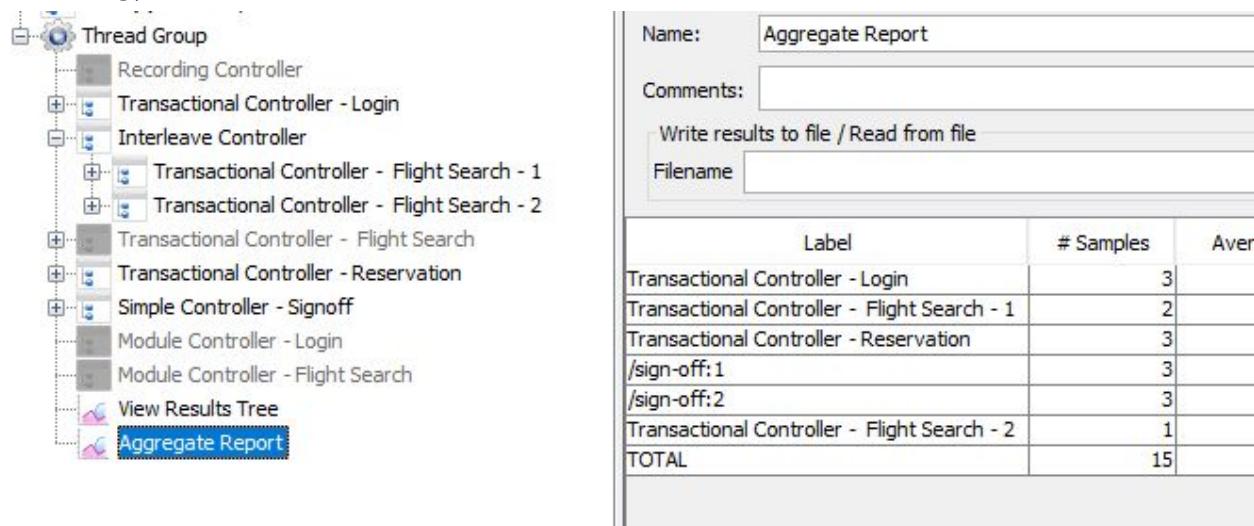
#### - Make Loop count - 2, and, then, run the project



Name:	Aggregate Report	
Comments:		
Write results to file / Read from file		
Filename		
Label	# Samples	Ave
Transactional Controller - Login	2	
Transactional Controller - Flight Search - 1	1	
Transactional Controller - Reservation	2	
/sign-off:1	2	
/sign-off:2	2	
Transactional Controller - Flight Search - 2	1	
TOTAL	10	

- Flight search-1 and 2 called only once - (even when Loop count was 2)
- Why? ...because Interleave Controller takes only 1.

- Loop count = 3, (FS 1 called twice and FS 2 called once)



#### NOTE:

- Interleave controller helps to handle Multiple search flows (Different search combinations)
-

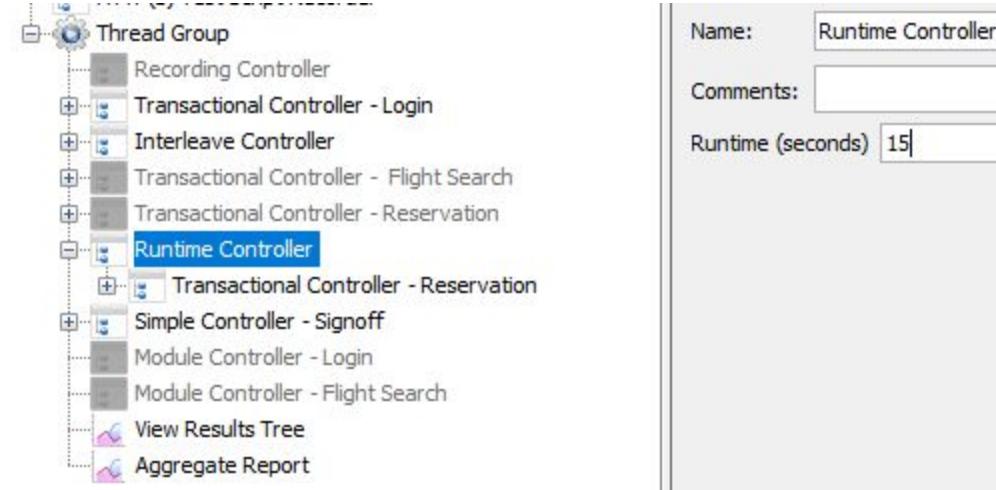
=====

**Study: Day-6 (July 02, 2020)**

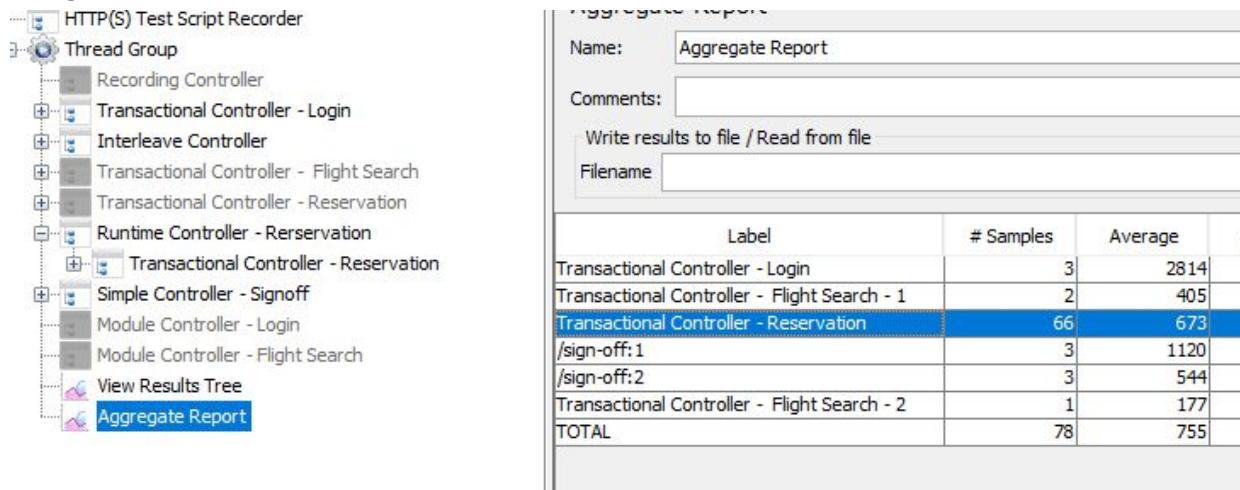
=====

## 6) Runtime Controller -

- Controls the execution of samplers/request for a given time
- MEANS -> Till how many seconds, this particular controller has to execute



- Run this controller for 15 seconds; for 15 seconds, this controller will keep on hitting the server with this Reservation samplers
- Example - if in 1 second, 2 samplers are getting hit, then, in 15 seconds, 30 samplers will hit the server, and then only, it will come out of this loop and goes to Sign Off.

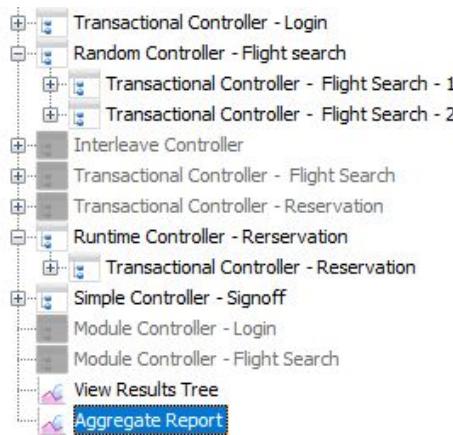


- Aggregate report -> samplers count for Reservation is much more than others (Login, Flight search, etc.)
- Now, where do you use this Runtime Controller in real time? -> has to be used where you have to rigorously put some load on server

- Example - **Black Friday sale** -> Payment gets failed sometimes;
  - Then, create a Runtime controller and put reservation samplers inside it
  - Run the reservation runtime controller only (no need to run others) for 5 min. or more and 100 threads or more
- 

## 7) Random Controller -

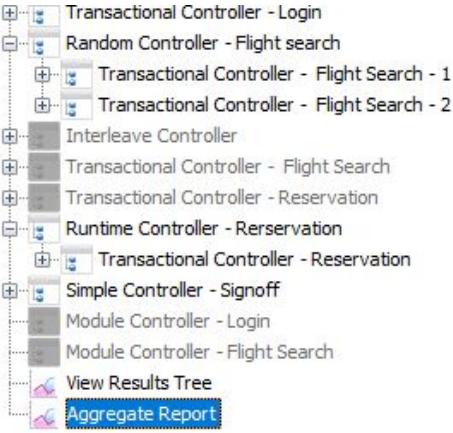
- Plays only one of its children samples picking it randomly
- Example - 3 samplers inside Random controller, then, it is not sure which one will be picked for iterations.
- It can be 3rd sampler in 1st iteration, 1st sampler in 2nd iteration or anything



Comments: Write results to file / Read from file  
Filename:

Label	# Samples	Average	Min	Max
Transactional Controller - Login	3	1489		
Transactional Controller - Flight Search - 2	2	179		
Transactional Controller - Reservation	15	579		
/sign-off:1	3	1015		
/sign-off:2	3	464		
Transactional Controller - Flight Search - 1	1	184		
TOTAL	27	671		

- Aggregate report -> **Flight search -2** is called 1st



Comments: Write results to file / Read from file  
Filename:

Label	# Samples	Average	Min	Max
Transactional Controller - Login	2	2337		
Transactional Controller - Flight Search - 1	1	171		
Transactional Controller - Reservation	9	632		
/sign-off:1	1	3350		
/sign-off:2	1	613		
Transactional Controller - Flight Search - 2	1	205		
TOTAL	15	980		

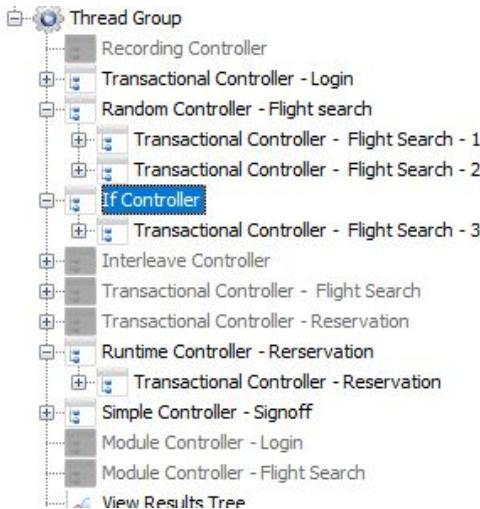
- Aggregate report -> **Flight search -1** is called 1st

## What is the difference between Random Controller and Random order controller?

- Random Order Controller plays all Samplers that are its children but in a random order
  - Random Controller plays only one of its children samples picking it randomly
-

## 8) If Controller -

- If the given condition is satisfied, then only, this controller will be executed by JMeter
- **True condition:** Flight search -3 will run



Name: If Controller

Comments:

**For performance it is advised to check "Interpret Condition as Variable Expression?" and use \_\_jexl3 or \_\_groovy evaluating to true or false or a variable.**

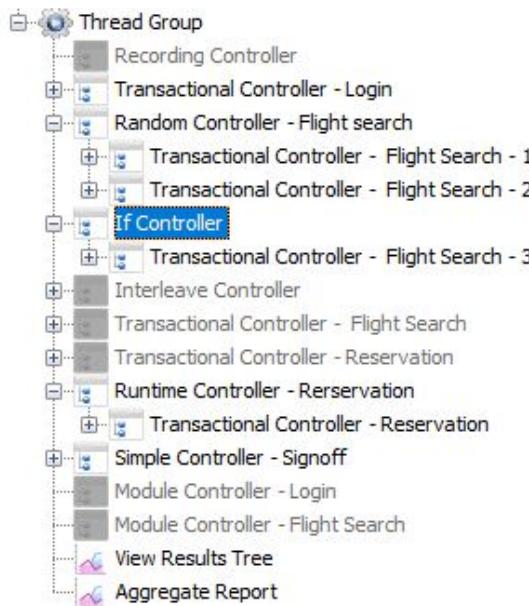
Expression (must evaluate to true or false)

Use status of last sample      \${JMETERThread.last\_sample\_ok} can be used here

Interpret Condition as Variable Expression?

Evaluate for all children?

- **False condition:** Flight search -3 will not run



Name: If Controller

Comments:

**For performance it is advised to check "Interpret Condition as Variable Expression?" and use \_\_jexl3 or \_\_groovy evaluating to true or false or a variable.**

Expression (must evaluate to true or false)

Use status of last sample      \${JMETERThread.last\_sample\_ok} can be used here

Interpret Condition as Variable Expression?

Evaluate for all children?

---

## Variables -

- You can add n no. of variables under Test plan, which can further be used in Thread Group
- Click on Test plan -> Add ->

The screenshot shows the JMeter Test Plan configuration. On the left, the Test Plan tree view includes nodes like HTTP(S) Test Script Recorder, Thread Group, Recording Controller, and various controllers like Transactional Controller, Random Controller, If Controller, and Interleave Controller. On the right, the 'Test Plan' dialog is open with the 'User Defined Variables' section. A table shows a single entry: 'myVariable' with a value of '10'. Below the table are buttons for Detail, Add, Add from Clipboard, Delete, Up, and Down.

- How to retrieve the value?

The screenshot shows the JMeter Test Plan configuration. The tree view includes nodes like Transactional Controller - Login, Random Controller - Flight search, and an 'If Controller' node which contains a 'Transactional Controller - Flight Search - 3' node. The 'If Controller' node has a condition 'Condition (evaluated with javascript)' set to '1 \${myVariable}==10'. A warning message at the top right says: 'For performance it is advised to check "Interpret Condition as and use \_\_jexl3 or \_\_groovy evaluating to true or false or a value"'. Below the condition, there are checkboxes for 'Use status of last sample' and 'Interpret Condition as Variable Expression?'.

- This Format is Javascript format .
- Aggregate report - Flight search -3 is executed (because condition is true).

The screenshot shows the JMeter Test Plan configuration and an 'Aggregate Report' table. The tree view includes nodes like Random Controller - Flight search, If Controller, and so on. The 'Aggregate Report' table on the right shows the following data:

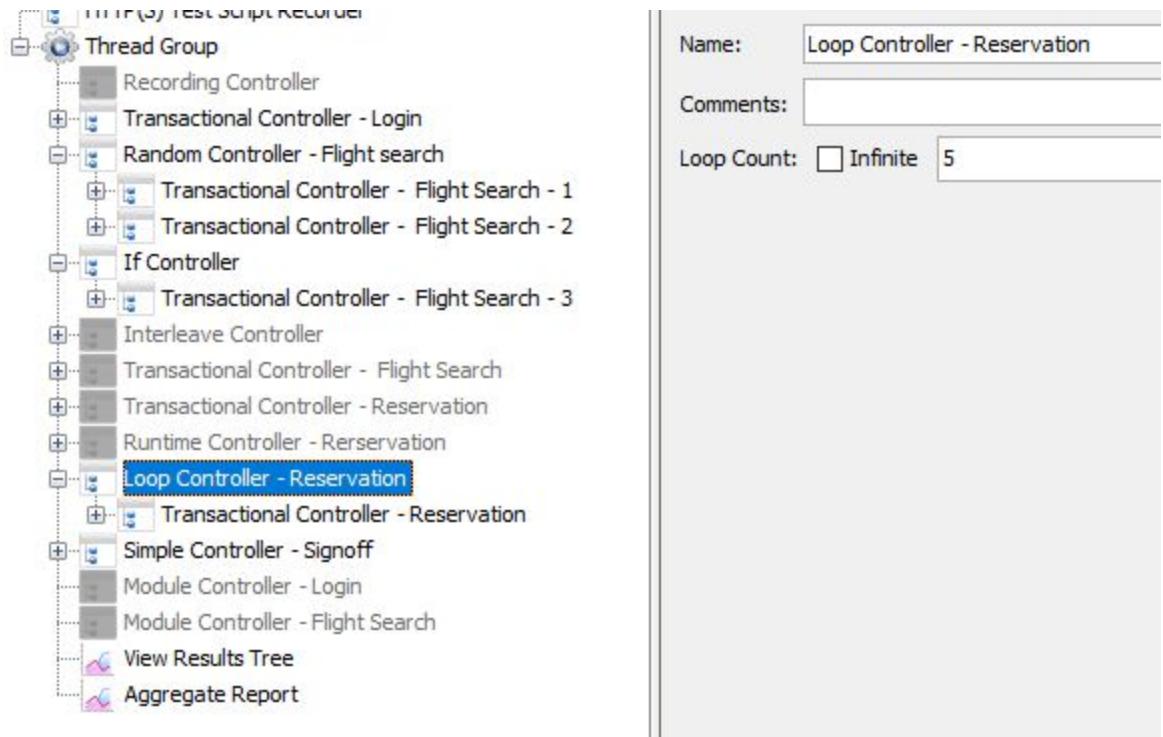
Label	# Samples	Average	Median
Transactional Controller - Login	2	4070	2465
Transactional Controller - Flight Search - 2	1	195	195
Transactional Controller - Flight Search - 3	2	189	189
Transactional Controller - Reservation	10	587	589
/sign-off:1	2	1049	397
/sign-off:2	2	718	693
Transactional Controller - Flight Search - 1	1	206	206
<b>TOTAL</b>	<b>20</b>	<b>916</b>	<b>589</b>

Now, what if you didn't define any variable in Test plan, but, used something in If Controller, then, what'll happen?

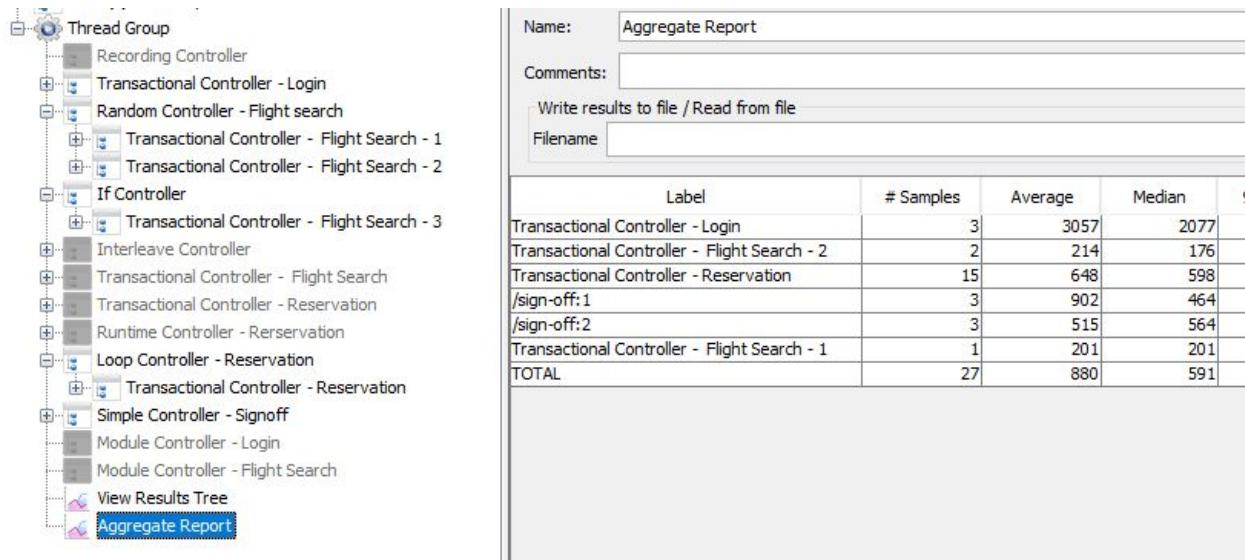
- No failures will be there; it'll take the condition as FALSE, and will execute the whole flow.

## 9) Loop Controller -

- Similar to Runtime Controller
- Runtime - Seconds has to be given (Time specific)
- Loop - how many times it should execute (no. of loops)
- 



- Aggregate report (For 3 users, this whole flow is working)-



=====

**Study: Day-7 (July 03, 2020)**

=====

=====

**9. Timers in Jmeter**

=====

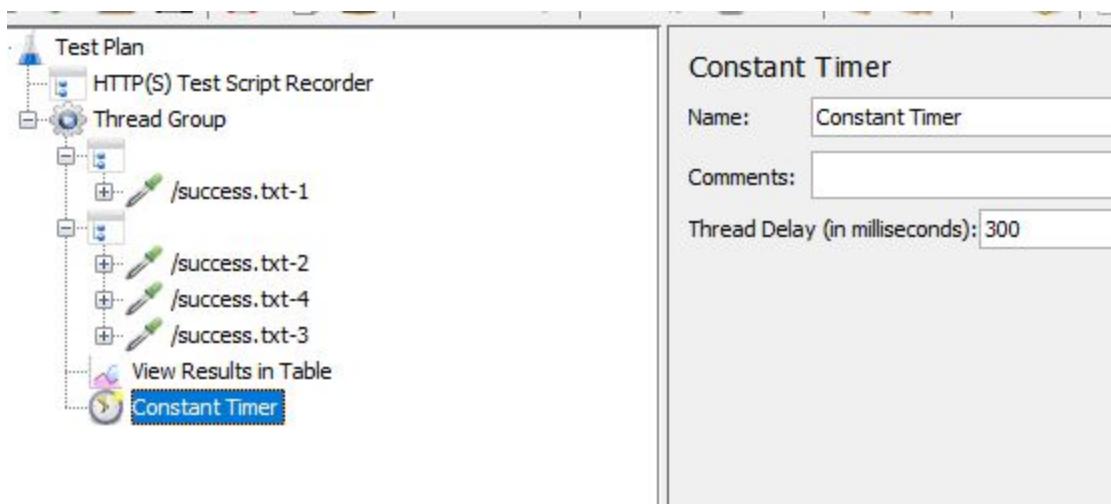
**Timers -**

- Not used inc actual load testing
- Example: 50 users ->
- Click on customer care link (in 1 or 2 min. May be 50 users will hit the URL, but, in 1 sec, all 50 Users will hit the same request is not not accurate behavior)
- We can call it as: Customer care link is clicked 50 times in 60 seconds
- Timers is simply Thread.sleep() in JAVA.

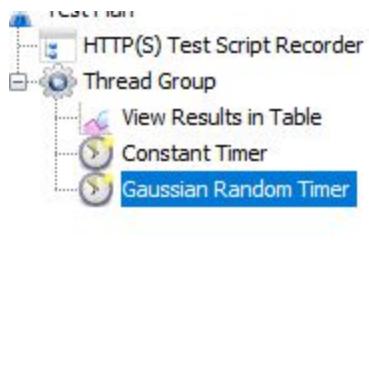
**Steps -**

- Do booking for any pair of flight and record it.
- Move samplers inside Thread Group
- Add Listener: View Results in Table
- Run the Project using 1 User and Loop count 10.
- Now, put timers in between the Loops (1st loop execution -> Timer (Sleep) -> 2nd loop execution)
- Right Click on Thread Group -> Add -> Timer ->

**1) Constant Timer -**



## 2) Gaussian Random Timer -



The screenshot shows the JMeter Test Plan tree on the left. A 'Gaussian Random Timer' node under the 'Thread Group' is highlighted with a blue selection bar.

**Gaussian Random Timer**

Name: Gaussian Random Timer

Comments:

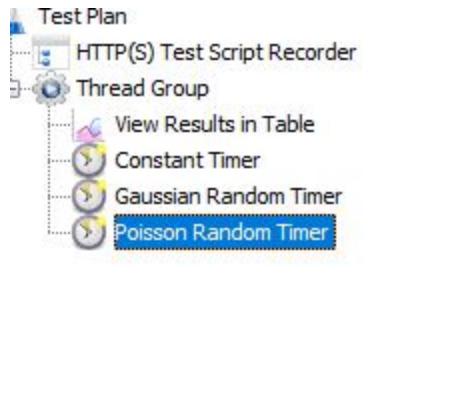
Thread Delay Properties

Deviation (in milliseconds):

Constant Delay Offset (in milliseconds):

- Deviation is the max. Time
- While execution, sleep will work for (300 + any value between 0-100)

## 3) Poisson Random Timer -



The screenshot shows the JMeter Test Plan tree on the left. A 'Poisson Random Timer' node under the 'Thread Group' is highlighted with a blue selection bar.

**Poisson Random Timer**

Name: Poisson Random Timer

Comments:

Thread Delay Properties

Lambda (in milliseconds):

Constant Delay Offset (in milliseconds):

- Lambda value will be generated by their algorithm

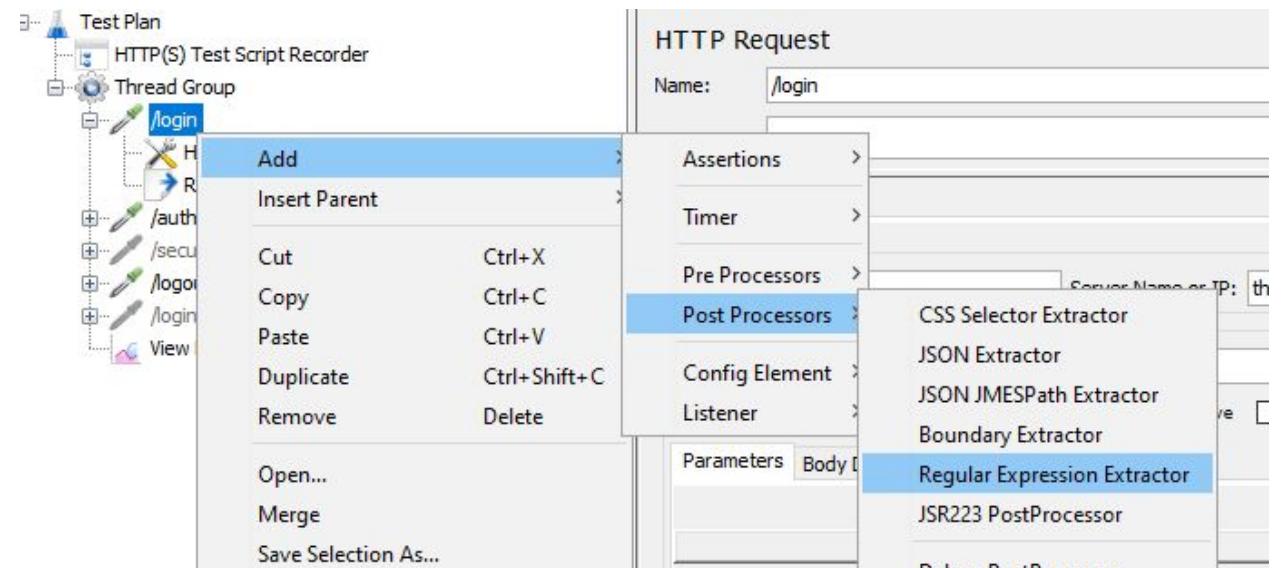
#### 4) Constant Throughput Timer -

The screenshot shows the JMeter Test Plan interface. On the left, under the 'Test Plan' tree, there is an 'HTTP(S) Test Script Recorder', a 'Thread Group' containing a 'View Results in Table' listener and three timers: 'Constant Timer', 'Gaussian Random Timer', 'Poisson Random Timer', and 'Constant Throughput Timer'. The 'Constant Throughput Timer' is selected and highlighted with a blue border. On the right, a detailed configuration dialog for the 'Constant Throughput Timer' is open. It has fields for 'Name' (set to 'Constant Throughput Timer'), 'Comments' (empty), 'Delay before each affected sampler' (empty), 'Target throughput (in samples per minute)' (set to '0,0'), and 'Calculate Throughput based on' (set to 'this thread only').

- Example -> Total servers : 4; maintaining 1 application
- 1 server - 100 requests per min
- Conclusion: 400 requests per min is the max capability
- 10 users -> 400 requests -> Requests have to be in queue

## =====10. Importance of Regular expressions in Jmeter=====

- Earlier, we did 1 recording (Heroku web app)
- We were giving the credentials (Username and password hard coded)
- This time, using REGEX concept, we will dynamically give these credentials
- Benefit of doing that - Suppose in future, if they change the username to tomcruise from tomsmith, even, then, our script won't fail.
- Because, we'll take these credentials value from their UI only.
- Right click on **/login** sampler -> Add ->Post Processors (means after performing all actions under /login sampler, this Post processor will start working)-> Regular Expression Extractor



Now, our **Goal** is to take value (username/password) from UI (which we get after hitting /login -> Response data) and use that value (username/password) to attach with /authenticate request.

---

## /login: -> Response data

Text

Sampler result Request Response data

Response Body Response headers

```
</div>
</div>
<div class="row">
<a href="https://github.com/touredave/the-internet"></a>
<div id="content" class="large-12 columns">
<div class="example">
<h2>Login Page</h2>
<h4>Subheader</h4>This is where you can log into the secure area. Enter <em>tomsmith</em> for the username and <em>SuperSecretPassword!</em> for the password. If the information is wrong you should see error messages.</h4>
<form name="login" id="login" action="/authenticate" method="post" >
<div class="row">
<div class="large-6 small-12 columns">
<label for="username">Username </label>
<input type="text" name="username" id="username" />
</div>
</div>
```

## /authenticate ->

- Previous:

Regular expression extractor

/authenticate

/secure

/logout

/login

View Results Tree

Basic Advanced

Web Server

Protocol [http]: https Server Name or IP: the-internet.herokuapp.com

HTTP Request

POST Path: /authenticate

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipart/form-data  Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?
username	tomsmith	<input type="checkbox"/>
password	SuperSecretPassword%21	<input type="checkbox"/>

- Now:

/authenticate

/secure

/logout

/login

View Results Tree

Basic Advanced

Web Server

Protocol [http]: https Server Name or IP: the-internet.herokuapp.com

HTTP Request

POST Path: /authenticate

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipart/form-data  Browser-compatible headers

Parameters Body Data Files Upload

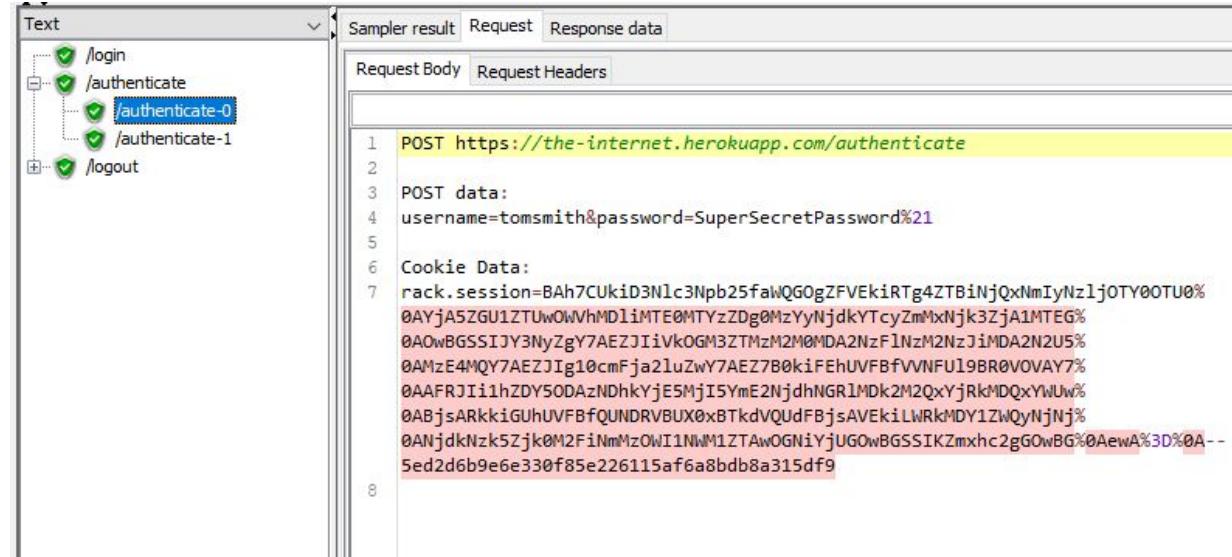
Send Parameters With the Request:

Name:	Value
username	\$(username_variable)
password	SuperSecretPassword%21

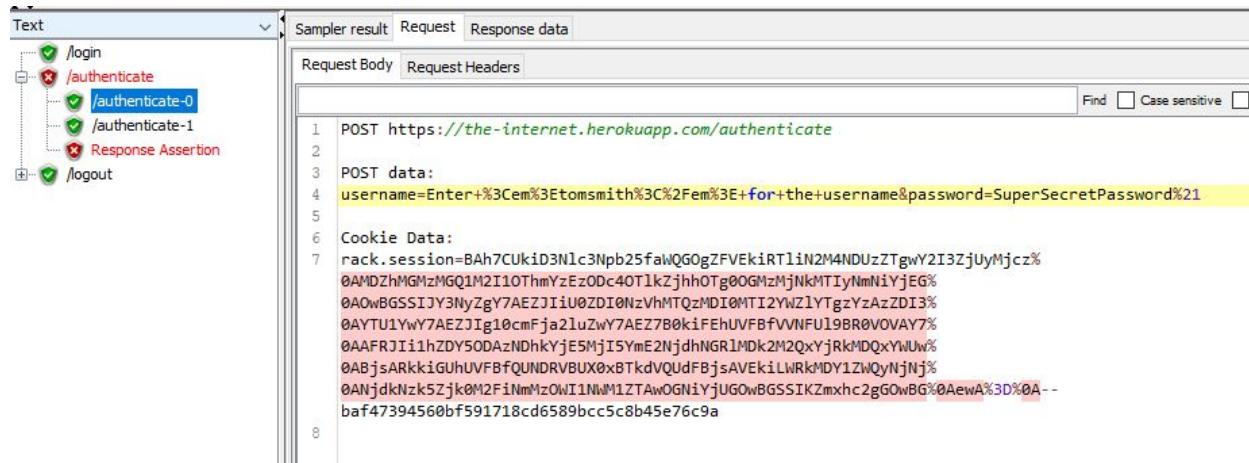
- Literals/Terminologies in the Regular Expression: <http://rexegg.com/>
- **Regular Expression:** Enter <em>(.+?)</em> for the username
- . -> Characters, + > any no. of characters, ? -> get the matching set from Response

The screenshot shows the JMeter interface. On the left, the 'Test Plan' tree view includes an 'HTTP(S) Test Script Recorder', a 'Thread Group' containing a 'login' step, an 'HTTP Header Manager', and a 'Regular Expression Extractor' step. Below these are steps for '/authenticate', '/secure', '/logout', and another '/login'. At the bottom is a 'View Results Tree' node. On the right, the 'Regular Expression Extractor' configuration dialog is open. It has fields for 'Name' (set to 'Regular Expression Extractor'), 'Comments', 'Apply to' (set to 'Main sample only'), 'Field to check' (set to 'Body'), 'Name of created variable' (set to 'username\_variable'), 'Regular Expression' (containing '(.+?)'), 'Template' (\$1\$), 'Match No.' (set to '0 for Random'), and 'Default Value' (set to 'admin'). There is also a checked checkbox 'Use em...

- **(.+?)** -> 1st group
- If there is any other (), then, it will be treated as 2nd group
- **\$1\$** -> 1st group value will be assigned to the variable (username\_variable)
- **\$0\$** -> **no group** (it will grab the entire value which you provided in the field: Name of created variable)
- Now, **what if 3 matches are found in the Response with our regular expression ?**
- Match No. (0 for Random): **1** (1st found pattern will be taken)
- Now, **what if no match is found in the Response with our regular expression ?**
- Default Value: **admin**
- **View Results Tree:**  
/authenticate-0 -> Request -> Request Body -> **Check username: tomsmith** has been passed



- If \$o\$ is given in Regular Expression extractor, see what value has passed as Username:



- If \$10\$ is given in Regular Expression extractor, then, null will be passed.

It's difficult to do all stuff manually, there is 1 concept called Debug Sampler  
(Right click on Thread Group -> Add -> Sampler -> Debug Sampler)

View Results Tree -> Debug Sampler (in last) -> Response data -> Response body

The screenshot shows the JMeter View Results Tree. On the left, there's a tree view of requests: Response Assertion, Response Assertion, Response Assertion, /secure, /logout, HTTP Header Manager, Response Assertion, Response Assertion, /login, View Results Tree, HTTP Cookie Manager, Assertion Results, and Debug Sampler. The 'View Results Tree' node is expanded. On the right, there's a 'Text' panel with a search bar and a 'Search' button. The 'Text' dropdown is set to 'Text'. The content area shows the response body of a 'Debug Sampler' request. It includes JMeter variables and their values:

```

JMeterVariables:
JMETERThread.last_sample_ok=true
JMETERThread.pack-org.apache.jmeter.threads.SamplePackage@30d116ef
START.HMS=123550
START.MS=1593759950213
START.YMD=20200703
TESTSTART.MS=1593765794193
__jm_Thread Group_idx=0
__jm_U_T_=Thread Group 1-1
__jmv_SAME_USER=true
username_variable=tomsmith
username_variable_g=1
username_variable_g0=Enter <em>tomsmith</em> for the username
username_variable_g1=tomsmith

```

- Add 1 more Regular Expression Extractor for PASSWORD, and check the output, how it looks in the DEBUG SAMPLER:

The screenshot shows the JMeter interface. On the left, the tree view includes an additional 'Regular Expression Extractor' under the '/authenticate' request. On the right, the 'Regular Expression Extractor' configuration dialog is open. It has sections for 'Comments', 'Apply to:' (set to 'Main sample only'), 'Field to check' (set to 'Body'), 'Name of created variable:' (set to 'password\_variable'), 'Regular Expression:' (set to 'and <em>(.)?</em> for the (password)'), 'Template (\$ $\$$  where  $i$  is capturing group number, starts at 1):' (set to '\$1'), 'Match No.' (set to '0 for Random'), and 'Default Value' (empty). Below the dialog, the 'View Results Tree' panel shows the 'username' and 'password' variables defined. On the far right, the 'Text' panel of a 'Debug Sampler' shows the expanded JMeter variables, including the newly extracted 'password\_variable'.

Name:	
username	$\$\{username\_variable\}$
password	$\$\{password\_variable\}$

```

__jm_U_T_=Thread Group 1-1
__jmv_SAME_USER=true
password_variable=SuperSecretPassword!
password_variable_g=2
password_variable_g0=and <em>SuperSecretPassword!</em> for the password
password_variable_g1=SuperSecretPassword!
password_variable_g2=password
username_variable=tomsmith
username_variable_g=1
username_variable_g0=Enter <em>tomsmith</em> for the username
username_variable_g1=tomsmith

```

Now, we created 2 different Regular Expression Extractors for 2 variables.

**Interview question:**

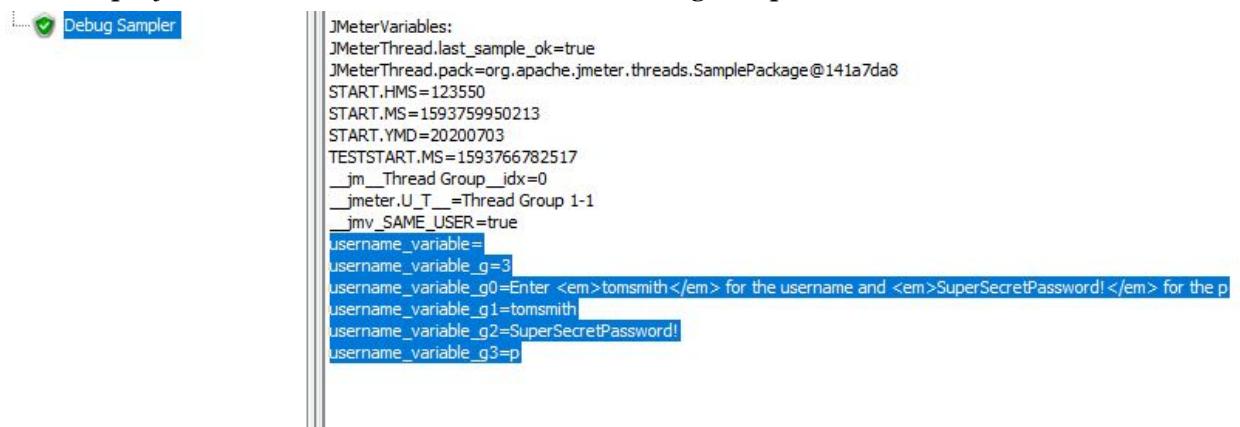
- How can you grab multiple variable using 1 Regular Expression Extractor?

There is a way: **Group Indexers**.

1) Regular Expression Extractor:

The screenshot shows the 'Regular Expression Extractor' configuration dialog. At the top, there are four radio buttons: 'Main sample and sub-samples' (unchecked), 'Main sample only' (checked), 'Sub-samples only' (unchecked), and 'JMeter variable name to use' (unchecked). Below this is a 'Field to check' dropdown with options: 'Body' (selected), 'Body (unescape)', 'Body as a Document', 'Response Headers', 'Request Headers', 'URL', and 'Request'. Under 'Name of created variable', the value 'username\_variable' is entered. In the 'Regular Expression' field, the placeholder text 'Enter <em>(.+?)</em> for the username and <em>(.+?)</em> for the (.+?)' is shown. The 'Template (\$\$ where i is capturing group number, starts at 1)' field is empty. The 'Match No. (0 for Random)' field contains '0'. The 'Default Value' field contains 'admin', and the 'Use empty default value' checkbox is unchecked.

2) Run the project and check View Results Tree- > Debug Sampler:



username\_variable\_g1=**tomsmith**, username\_variable\_g2=**SuperSecretPassword!**

G stands for **Global**.

3) /authenticate:

The screenshot shows the 'Parameters' tab for a sampler. At the top, there are three tabs: 'Parameters' (selected), 'Body Data', and 'Files Upload'. Below the tabs, a message says 'Send Parameters With the Request:'. A table lists parameters with their names, values, and URL encoding status. The table has columns for 'Name', 'Value', and 'URL Encode'. There are two rows: one for 'username' with value '\${username\_variable\_g1}' and 'URL Encode' checked, and one for 'password' with value '\${username\_variable\_g2}' and 'URL Encode' checked.

Name:	Value	URL Encode
username	\${username_variable_g1}	<input checked="" type="checkbox"/>
password	\${username_variable_g2}	<input checked="" type="checkbox"/>

**NOTE:** You can check the Regular Expression Extractor from View Results Tree as well.

- Regex\_Working\_2 -> Click on Drop down ->

The screenshot shows the JMeter interface with a Test Plan tree on the left and a View Results Tree window on the right.

**Test Plan Tree:**

- Test Plan
  - HTTP(S) Test Script Recorder
  - Thread Group
    - /login
      - Regular Expression Extractor
      - Regular Expression Extractor
      - HTTP Header Manager
      - Regular Expression Extractor
    - /authenticate
      - Regular Expression Extractor
    - /secure
    - /logout
    - /login
  - View Results Tree
  - HTTP Cookie Manager
  - Assertion Results
  - Debug Sampler

**View Results Tree Window:**

- Name:** View Results Tree
- Comments:** Write results to file / Read from file
- Filename:** [Browse...]
- Search:** [Search] [Reset]
- RegEx Tester:** Shows a tree structure of regular expression extractors under various URLs: /login, /authenticate, /logout, and Debug Sampler.
- Content:** Displays the HTML response body of a page, showing DOCTYPE, head, and body sections.
- Regular expression:** Enter <em>(.+?)</em> for the username
- Results:** Match count: 1  
Match[1][0]=Enter <em>tomsmith</em> for the username  
Match[1][1]=tomsmith

=====  
===== **Study: Day-8 (July 05, 2020)** =====  
=====

===== **11. Data Driven testing with Jmeter** =====

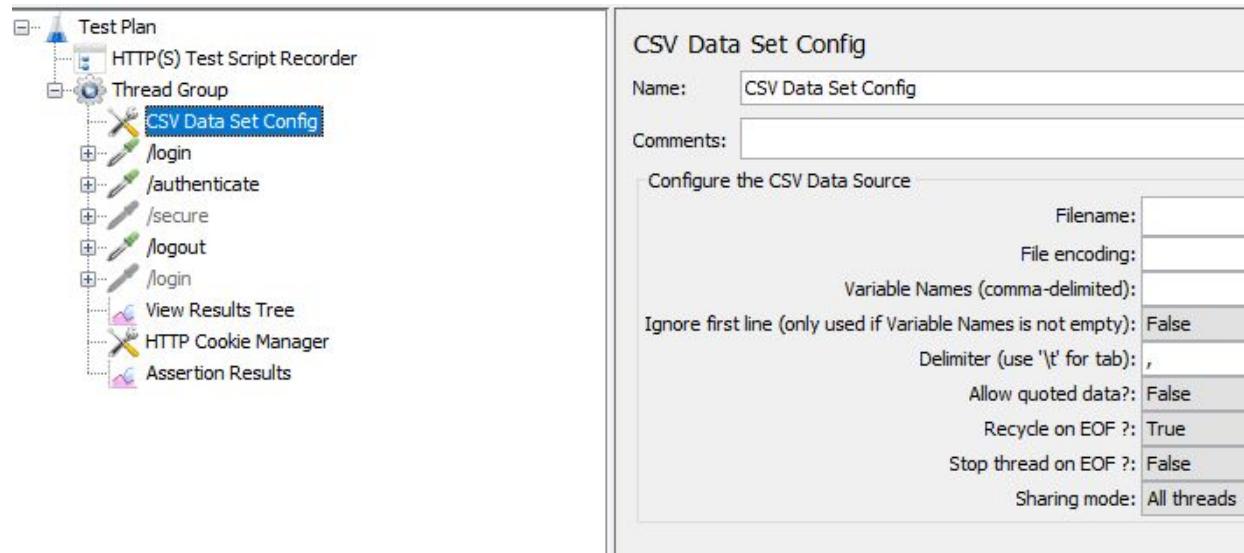
**Project for reference: Data\_Driven\_2.jmx**

- Do Registration (Sign up) on WebTours and record the Scripts.
- We don't want to hardcode the values; want to get the values from some external source (in 1 script run, 10 users' will be registered)

**NOTE:** Right now, I'm unable to do new recording in JMeter, so I'm using the previously prepared scripts.

**Steps -**

- Right click on Thread Group -> Add -> Config Element -> CSV Data Set Config



- Variable Names (comma-delimited) - username,password

```
Untitled - Notepad
File Edit Format View Help
tomsmith,SuperSecretPassword%21
rajatt95,rajatt95
user,pass
tomsmith,SuperSecretPassword%21
```

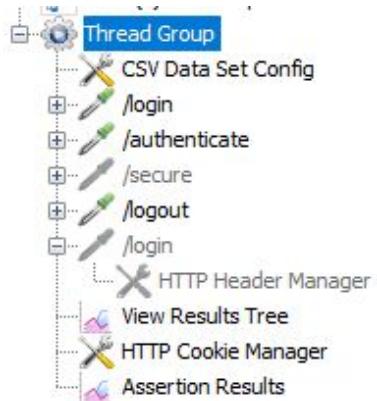
- Notepad ->

- Save this file as: data.csv (Location: D:\Work in Local machine\Projects\Learning\Learn JMETER from Scratch on Live Apps -Performance Testing)
- **Recycle on EOF:** true (Project ran for 100 Users and, only 7 Users entries are present in data.csv file, so for 8th user, 1st data set will be taken). If it is **false**, then, 7th data set will be last. Script won't take any data and stop the script.
- **EOF:** End of File

The screenshot shows the JMeter interface with a test plan containing the following elements:

- CSV Data Set Config:** A configuration element under a Thread Group. It has the following settings:
  - Name: CSV Data Set Config
  - Comments:
  - Configure the CSV Data Source:
    - Filename: D:/Work in Local machine/Projects/Learning/Learn JMETER from Scratch on Live Apps -Performance Testing/data.csv
    - File encoding:
    - Variable Names (comma-delimited): datadriven\_username,datadriven\_password
    - Ignore first line (only used if Variable Names is not empty): False
    - Delimiter (use '\t' for tab): ,
    - Allow quoted data?: False
    - Recycle on EOF ?: True
    - Stop thread on EOF ?: False
    - Sharing mode: Current thread
- Thread Group:** Contains the CSV Data Set Config and several other samplers:
  - /authenticate
  - /login
  - /secure
  - /logout
  - /login
- HTTP Request:** Configuration for the /authenticate sampler.
  - Name: /authenticate
  - Comments: Detected the start of a redirect chain
  - Basic tab selected:
    - Web Server:
      - Protocol [http]: https
      - Server Name or IP: the-internet.herokuapp.com
    - HTTP Request:
      - Method: POST
      - Path: /authenticate
      - Checkboxes: Redirect Automatically (unchecked), Follow Redirects (checked), Use KeepAlive (checked), Use multipart/form-data (unchecked), Browser-compatible headers (unchecked)
    - Parameters tab:
      - username: \${datadriven\_username}
      - password: \${datadriven\_password}
  - Advanced tab (disabled): Send Parameters With the Request
- View Results Tree**
- Assertion Results**

- 1st Thread will take 1st data set, 2nd thread will go for 2nd data set.



Name: Thread Group

Comments:

Action to be taken after a Sampler error

Continue    Start Next Thread Loop    Stop Thread

#### Thread Properties

Number of Threads (users):

Ramp-up period (seconds):

Loop Count:  Infinite

Same user on each iteration

Delay Thread creation until needed

Specify Thread lifetime

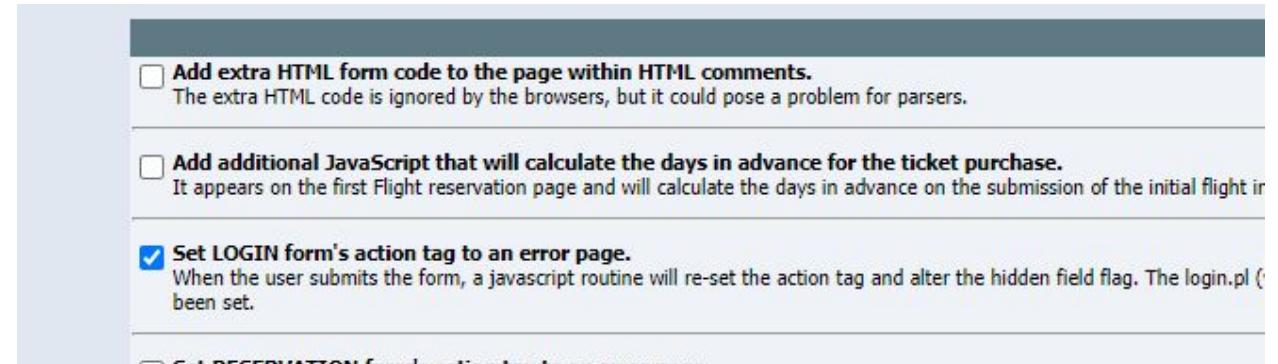
Duration (seconds):

Startup delay (seconds):

## =====12. Handling Dynamic Responses -Advanced Jmeter-2=====

### WebTours application -

- Enable the checkbox (you'll get this webpage after click on Administration link); to enable the Session ID concept in the WebTours application



- Record Login-Logout operation on Webtours application.
- Save the Project: "Correlation".
- Don't just think that everything is ok after seeing green tick icons in View Results Tree

- A User session ID has been generated after successful login.
- That session ID validity is dependent on Developer (can be of 1 min., 1 hour, 1 day or till the User closes the browser window)
- Session ID is always dynamic
- Login functionality always has session ID stuff.
- How to handle it? - **Correlation**
- **Correlation -**
  - Dynamically handle this session ID value
  - Take value from previous response and add that value in the next request.
  - 2nd response -> 3rd request

Last time, we used HTTP Cookie Manager to handle this kind of situation.

### Tutor project: 4. Regular expressions code.jmx

- In this Project, What Tutor has done? - He is storing the value of session id in 1 variable (Post processors -> Regular Expression Extractor : **session\_id**) and using that variable in the next request(**userSession: \${session\_id}** ).

**Test Plan**

- Correlation
  - 8 /cgi-bin/welcome.pl
  - 8 /cgi-bin/welcome.pl
  - 9 /cgi-bin/nav.pl
  - HTTP Header Manager
    - Regular Expression Extractor
  - 10 /cgi-bin/login.pl
  - 10 /cgi-bin/login.pl
  - 11 /cgi-bin/login.pl
  - 12 /cgi-bin/nav.pl
  - 13 /cgi-bin/welcome.pl
- View Results Tree
- WorkBench

**Regular Expression Extractor**

Name: Regular Expression Extractor

Comments:

Apply to:

Main sample and sub-samples  Main sample only  Sub-samples

Field to check:

Body  Body (unesCAPed)  Body as a Document  R

Reference Name: sessionid

Regular Expression: name="userSession" value=(.+?)>

Template: \$1

Match No. (0 for Random): 1

Default Value:

**Test Plan**

- Correlation
  - 8 /cgi-bin/welcome.pl
  - 8 /cgi-bin/welcome.pl
  - 9 /cgi-bin/nav.pl
  - HTTP Header Manager
    - Regular Expression Extractor
  - 10 /cgi-bin/login.pl
  - 10 /cgi-bin/login.pl
  - 11 /cgi-bin/login.pl
  - 12 /cgi-bin/nav.pl
  - 13 /cgi-bin/welcome.pl
  - Debug Sampler
- View Results Tree
- WorkBench

**HTTP Request**

Name: 10 /cgi-bin/login.pl

Comments:

Web Server

Server Name or IP: 192.168.56.1 port Number: 1080 Connect: R

Timeouts (milliseconds)

HTTP Request

Implementation: Protocol [http]: http Method: POST Content encoding:

Path: /cgi-bin/login.pl

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipartform-data for POST  Browser-compatible headers

**Parameters** **Body Data**

Send Parameters With the Request:	
Name:	Value
userSession	\$1(sessionid)
username	rahul
password	meter

Detail Add Add from Clipboard Delete Up Down

**Send Files With the Request:**

File Path: P

## =====13. Jmeter validations in Non GUI Mode=====

### Non GUI mode -

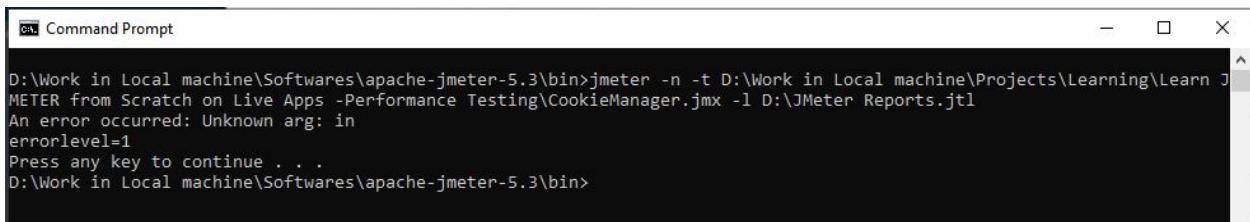
- No need to open any tool (Earlier we were using GUI mode; ApacheJMeter.jar)

### Steps -

- Open command prompt in administrative mode
- Command to navigate to bin directory: **cd D:\Work in Local machine\Softwares\apache-jmeter-5.3\bin**
- Command to run the Test plan: **jmeter -n -t D:\Work in Local machine\Projects\Learning\Learn JMETER from Scratch on Live Apps -Performance Testing\CookieManager.jmx**
- **-n:**Non GUI mode, **-t:** Test plan
- But, how will you see the results?
- Command to run the Test plan with Reporting: **jmeter -n -t D:\Work in Local machine\Projects\Learning\Learn JMETER from Scratch on Live Apps -Performance Testing\CookieManager.jmx -l**
- **-l:** Reporting
- But, where will you store the reports?
- For that, you need to add the path in the command

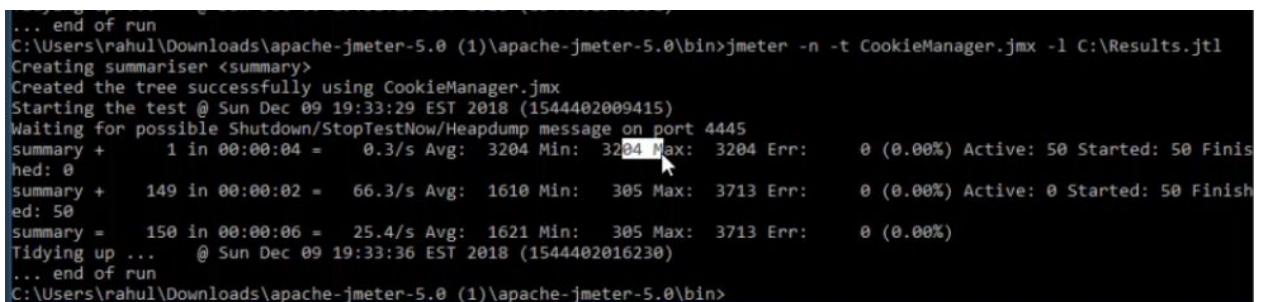
### Final command:

- Command to run the Test plan with Reporting: **jmeter -n -t D:\Work in Local machine\Projects\Learning\Learn JMETER from Scratch on Live Apps -Performance Testing\CookieManager.jmx -l D:\JMeter Reports.jtl**
- **JMeter Reports.jtl:** File name
- If you will run this command in no-administrative mode, then,



The screenshot shows a Command Prompt window with the following text:  
D:\Work in Local machine\Softwares\apache-jmeter-5.3\bin>jmeter -n -t D:\Work in Local machine\Projects\Learning\Learn JMETER from Scratch on Live Apps -Performance Testing\CookieManager.jmx -l D:\JMeter Reports.jtl  
An error occurred: Unknown arg: in  
errorlevel=1  
Press any key to continue . . .  
D:\Work in Local machine\Softwares\apache-jmeter-5.3\bin>

- And, in administrative mode:



The screenshot shows a Command Prompt window with the following output:  
... end of run  
C:\Users\rahul\Downloads\apache-jmeter-5.0 (1)\apache-jmeter-5.0\bin>jmeter -n -t CookieManager.jmx -l C:\Results.jtl  
Creating summariser <summary>  
Created the tree successfully using CookieManager.jmx  
Starting the test @ Sun Dec 09 19:33:29 EST 2018 (1544402009415)  
Waiting for possible Shutdown/StopTestNow/Heapdump message on port 4445  
summary + 1 in 00:00:04 = 0.3/s Avg: 3204 Min: 3204 Max: 3204 Err: 0 (0.00%) Active: 50 Started: 50 Finished: 0  
summary + 149 in 00:00:02 = 66.3/s Avg: 1610 Min: 305 Max: 3713 Err: 0 (0.00%) Active: 0 Started: 50 Finished: 50  
summary = 150 in 00:00:06 = 25.4/s Avg: 1621 Min: 305 Max: 3713 Err: 0 (0.00%)  
Tidying up ... @ Sun Dec 09 19:33:36 EST 2018 (1544402016230)  
... end of run  
C:\Users\rahul\Downloads\apache-jmeter-5.0 (1)\apache-jmeter-5.0\bin>

- You'll be able to see no. of parameters (Average, Min. time, Max. time, etc.)
- Detailed report:
  - Go to your mentioned path(**D:\JMeter Reports.jtl**):
  - You need to drag and drop this report in Excel to understand (Looks weird)

```

ip,elapsed,label,responseCode,responseMessage,threadName,dataType,success,failureMessage,bytes,sentBytes,grpThreads,allThreads,URL,Latency,IdleTime
47108,1690,8 /login,200,OK,Thread Group 1-1,text,true,,3213,345,1,1,https://the-internet.herokuapp.com/login,1690,0,1626
48803,100,13 /authenticate,200,OK,Thread Group 1-1,text,true,,3846,2099,1,1,https://the-internet.herokuapp.com/secure,51,0
48803,51,13 /authenticate-0,303,See Other,Thread Group 1-1,text,true,,995,1037,1,1,https://the-internet.herokuapp.com/authenticate,51,0,0
48857,46,13 /authenticate-1,200,OK,Thread Group 1-1,text,true,,2851,1062,1,1,https://the-internet.herokuapp.com/secure,46,0
48904,81,18 /logout,200,OK,Thread Group 1-1,text,true,,4358,1938,1,1,https://the-internet.herokuapp.com/login,43,0,0
48904,43,18 /logout-0,302,Found,Thread Group 1-1,text,true,,962,953,1,1,https://the-internet.herokuapp.com/logout,43,0,0
48949,36,18 /logout-1,200,OK,Thread Group 1-1,text,true,,3396,985,1,1,https://the-internet.herokuapp.com/login,36,0
11158,2801,8 /login,200,OK,Thread Group 1-30,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,2801,0,2707
10770,3204,8 /login,200,OK,Thread Group 1-9,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,3204,0,3117
10749,3204,8 /login,200,OK,Thread Group 1-1,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,3204,0,3115
10743,3203,8 /login,200,OK,Thread Group 1-7,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,3203,0,3134
11122,2824,8 /login,200,OK,Thread Group 1-27,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,2823,0,2757
10941,3068,8 /login,200,OK,Thread Group 1-20,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,3068,0,2933
11074,2971,8 /login,200,OK,Thread Group 1-26,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,2971,0,2817
11362,2700,8 /login,200,OK,Thread Group 1-40,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,2700,0,2542
10945,3120,8 /login,200,OK,Thread Group 1-21,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,3120,0,2945
10907,3185,8 /login,200,OK,Thread Group 1-19,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,3185,0,2982
10858,3244,8 /login,200,OK,Thread Group 1-16,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,3244,0,3046
11308,2836,8 /login,200,OK,Thread Group 1-37,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,2836,0,2600
11463,2694,8 /login,200,OK,Thread Group 1-44,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,2694,0,2393
11223,2938,8 /login,200,OK,Thread Group 1-33,text,true,,3213,345,50,50,https://the-internet.herokuapp.com/login,2938,0,2684

```

11:10  
Count: 10

## External tools to explore the Report -

- 1) Taurus - Framework
- 2) Blazemeter Cloud (Upload the file to their server and get your report generated)

---

## Steps -

- Go to <https://a.blazemeter.com>
- Click on Create test

The screenshot shows the BlazeMeter web interface. At the top, there is a navigation bar with a dropdown menu labeled 'orts' and a circular icon containing the number '0'. To the right of the navigation bar is a blue 'Create Test' button. Below the navigation bar, the main content area has a heading 'Select a test type'. Underneath this, there is a sub-section titled 'Performance Tests' with a purple button labeled 'Performance Test'. To the left of the 'Performance Test' button, there is a descriptive text: 'Upload JMeter, Gatling, The Grinder, Taurus YAML and more, or input a list of URLs/API Endpoints. Use Multi Test to combine tests into a single execution and consolidated report.' Below the 'Performance Tests' section, there is another section titled 'Test Creation Tools' with a yellow 'Get it now' button. To the left of the 'Get it now' button, there is a sub-section with the text: 'You can easily create a Taurus YAML Selenium script with the BlazeMeter Chrome extension.' Below this, there is another sub-section with the text: 'Create tests by recording via proxy' and a red 'Recorder' button.

orts ▾ 0

Create Test

Select a test type

This test will be created in project: Default project ▾

## Performance Tests

Upload JMeter, Gatling, The Grinder, Taurus YAML and more, or input a list of URLs/API Endpoints. Use Multi Test to combine tests into a single execution and consolidated report.

Performance Test

## Test Creation Tools

You can easily create a Taurus YAML Selenium script with the BlazeMeter Chrome extension.

Get it now

Create tests by recording via proxy

Recorder

- Then, click on Performance Test (we already have scripts ready with us, so, no need to go for Test Creation Tools)

The screenshot shows the Blazemeter configuration interface. At the top, there are navigation links for 'Tests' (with 0 results), 'Create Test', and tabs for 'Configuration' (selected) and 'History'. A large blue '+' button is centered above the configuration area. Below it, the text 'Upload Script' with a help icon is displayed, followed by instructions: 'Test with an existing script or use a Shared Folder. We support JMeter, Gatling, Selenium, Taurus and many more.' On the left, a sidebar shows a file structure with a .dd file selected, and a progress bar at 0%. The main configuration section is titled 'LOAD CONFIGURATION' with a help icon. It includes sliders for 'Total Users' (set to 20), 'Duration (min)' (set to 1), and 'Ramp Up Time (min)' (set to 1). A 'Get More' link is also present. To the right, there's a preview of the test results showing a single blue line graph.

- You can put Load using Blazemeter as well.
- It's foolishness to run the Test plan using your own machine.
- Run the test by doing your required configurations

## Launch Test

Your test will run with following configuration:

- ✓ 1 Engine
- ✓ 20 Total threads
- ✓ Max duration of 1 minute

Click **Launch Servers** to start the servers and run the test. The test will start as soon as the servers are ready.

Run test in the background

**Launch Servers**

**BlazeMeter** Functional Performance Mock Services API Monitoring Rajat Verma

Home Default workspace Projects Tests Reports 1 Create Test

Default project /July\_05\_4:14 PM Scenario: July\_05\_4:14 PM Location: US East (Virginia, Google)

July\_05\_4:14 PM ...

Summary Timeline Report Request Stats Engine Health Errors Logs Original Test Configuration

**20 VU**  
Max Users

**133.5 Hits/s**  
Avg. Throughput

**0.02 %**  
Errors

**150.02 ms**  
Avg. Response Time

**249 ms**  
90% Response Time

**472.8 KiB/s**  
Avg. Bandwidth

⌚ Duration: 1 minute

▶ Started: Jul 05, 2020, 4:22:54 PM

🕒 ETA: Any time now..

⌚ Test Type: JMeter

📍 Locations: US East (Virginia, Google)

🌐 Response Codes: 2xx

笔记:

Save Note

**Load**

**Response Time**

Default project /July\_05\_4:14 PM Scenario: July\_05\_4:14 PM Location: US East (Virginia, Google)

July\_05\_4:14 PM ...

Summary Timeline Report Request Stats Engine Health Errors Logs Original Test Configuration

**Load**

**Response Time**

Filter By Label:  Results per page: 25 Go to page: 1

Results: 4 out of 4 labels

Element Label...	# Samples ▲▼	Avg. Response Ti...	Avg. Hits/s ▲▼	90% line (ms) ▲▼	95% line (ms) ▲▼	99% line (ms) ▲▼	Min Response Ti...	Max Response Ti...	Avg. Bandwidth (...)	Error Percentag...
ALL	8260	144.07	137.67	236	289	415	13	4191	487.65	0.01%
/authenticate	2756	130.78	46.71	226	263	341	16	3279	175.44	0.04%
/logout	2748	122.82	46.58	202	236	309	13	999	198.22	0%
/login	2756	178.7	45.93	295	369	1247	35	4191	120.21	0%

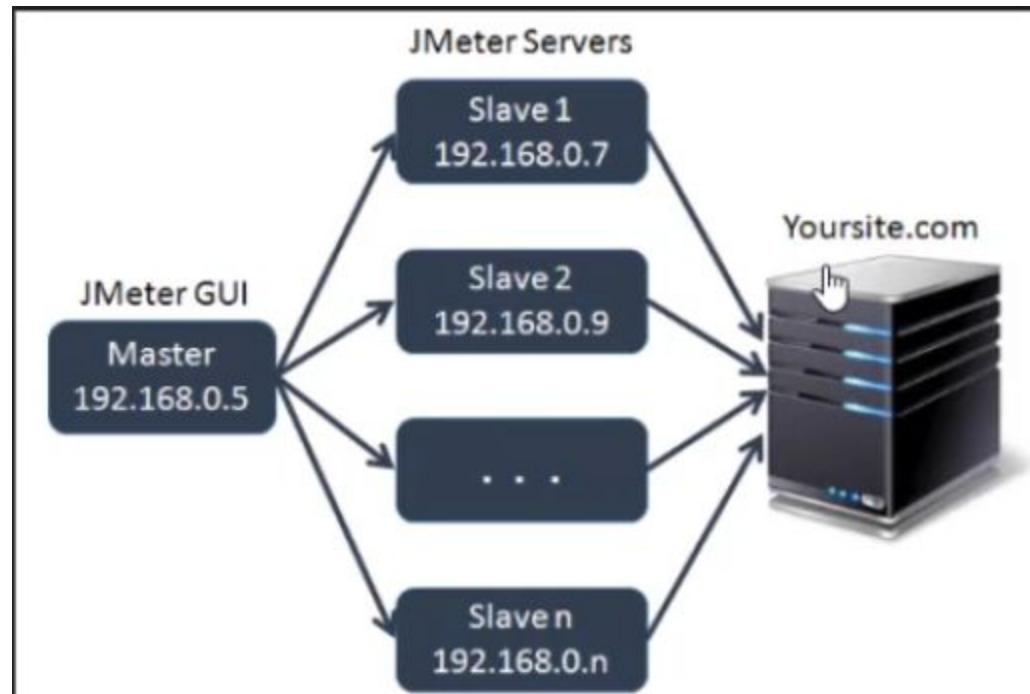
=====Study: Day-9 (July 06, 2020)=====

=====14. Jmeter Distributed Mode in running tests in Slave machines=====

- **Recommendation -**

- We should not never run load test on our own machine because we have limited CPU configuration (RAM, storage.....)
- It becomes Restriction by the machine it self when we run the load test using many users (example: 5000 users)
- Because of that we may get the wrong outputs as well.

- **Architecture of JMeter Distributed Mode -**



- We need to do the configurations in JMeter in such a way that my main machine will be the master and my other machines will be the Slave.
- Example - Running the test for 5000 Users and I do have 5 machines other than the machine machine, then, I'll give 1000 Users per machine(slaves).
- We have brought the Traffic from different machines (Real Time scenario)
- Benefit of doing that -
  - We will get the actual results.

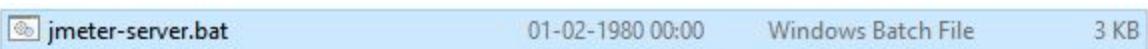
## Steps to start JMeter Slave server -

Take 2 machines (Make your machine as slave and other one as Master)

**My machine ->**

- D:\Work in Local machine\Softwares\apache-jmeter-5.3\bin -> jmeter.properties
- #server.rmi.ssl.disable=false -> server.rmi.ssl.disable=true

```
# Set this if you don't want to use SSL for RMI
#server.rmi.ssl.disable=false
server.rmi.ssl.disable=true
```
- Start “jmeter-server” :



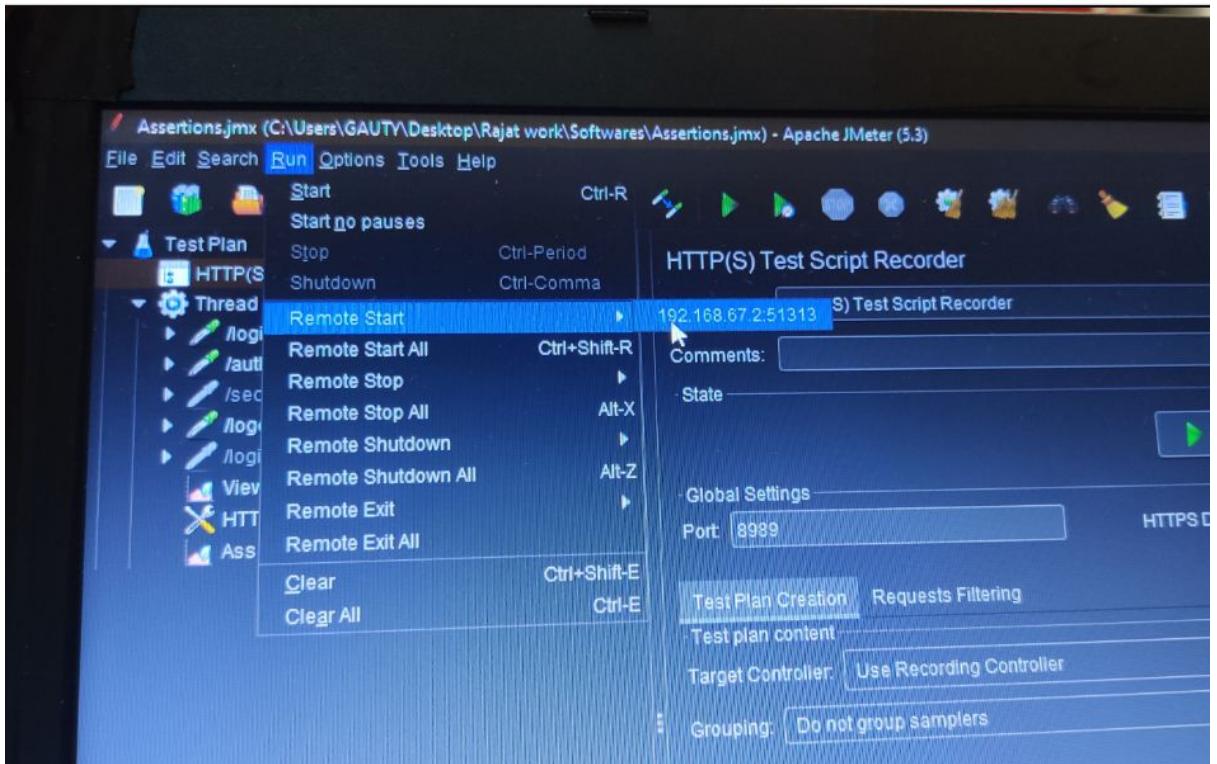
The screenshot shows a Windows File Explorer window with a single item: "jmeter-server.bat". Below the file name, it displays the last modified date and time as "01-02-1980 00:00", the file type as "Windows Batch File", and the size as "3 KB". The file content is visible in a text area below the header:  
Could not find ApacheJMeter\_core.jar ...  
... Trying JMETER\_HOME=..  
Found ApacheJMeter\_core.jar  
Created remote object: UnicastServerRef2 [liveRef: [endpoint:[192.168.67.2:51313](local),objID: [-4ef3cd37:1732238b03b:-7fff, 1750050848889134600]]]

- 
- IP address: **192.168.67.2:51313**

## Steps to start JMeter MASTER server -

- Download JMeter in Master machine (same version)
- Open the jmeter.properties file (inside bin directory)
- #server.rmi.ssl.disable=false -> server.rmi.ssl.disable=true

```
# Set this if you don't want to use SSL for RMI
#server.rmi.ssl.disable=false
server.rmi.ssl.disable=true
```
- remote\_hosts=127.0.0.1 (by default, Local machine IP is given) ->  
remote\_hosts=**192.168.67.2**
- And, if you want to connect more slaves, then, give those slave's IP addresses by using comma as a Delimiter
- Open JMeter and Run -> Remote Start -> Select the IP address of Slave



Now, this test plan will run in the slave machine (means Traffic has diverted in Slave machine).

- You will still be able to see results in the Master machine, but, execution will go in Slave machine.

### Slave machine Command Prompt -

```

Thread Group
Name: Thread Group

Select C:\WINDOWS\system32\cmd.exe
Could not find ApacheJmeter_core.jar ...
... Trying JMETER_HOME=...
Found ApacheJmeter_core.jar
Created remote object: Uni
ff, 732868511349243614]]
Starting the test on host
Finished the test on host

```

- Test plan execution started

### Important points to keep in mind:

- Same JMeter version, Same sub-network
- remote\_hosts requires only the address
- server\_port - To specify the Port (Optional)

=====15. Beanshell Scripting basics for Jmeter=====

**Beanshell** is an Scripting language

## **Steps -**

- Thread Group -> Add -> Sampler -> BeanShell Sampler
  - Options -> Log Viewer

```
4 var a=12;
5 var b=12;
6
7 var sum=a+b;
8
9 log.info("Sum: "+sum);
10
11 //String parameter has to be passed in Log.info()
12 //log.info(sum); - This will give Error - Error invoking bsh method
13
14 //Putting Hi in variable a (Earlier a had 12, now, will have Hi)
15 vars.put("a","Hi");
16 log.info("After Hi: "+vars.get("a"));
17
18 vars.put("a","Bye");
19 log.info("After Bye: "+vars.get("a"));
20
21 //ctx - JMeter Context variable : You get the Thread no. on which your test plan is
22 log.info("Thread number:"+ctx.getThreadNum());
23

2020-07-06 10:33:51,432 INFO o.a.j.u.BeanShellTestElement: After Hi: Hi
2020-07-06 10:33:51,432 INFO o.a.j.u.BeanShellTestElement: After Bye: Bye
2020-07-06 10:33:51,432 INFO o.a.j.u.BeanShellTestElement: Thread number:0
2020-07-06 10:33:51,433 INFO o.a.j.t.JMeterThread: Thread is done: Thread Group 1-1
2020-07-06 10:33:51,433 INFO o.a.j.t.JMeterThread: Thread finished: Thread Group 1-1
2020-07-06 10:33:51,433 INFO o.a.j.e.StandardJMeterEngine: Notifying test listeners of end of test
2020-07-06 10:33:51,433 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)
```

## What our goal is to do Signup for 10 Users using BeanShell scripting, everytime different Username has to be passed?

- Before the Test plan execution starts, using BeanShell scripting, we will change the Username.

Test Plan

Name:

Comments:

User Defined Variables

Name:	
userId	bean

```
int threadNum=ctx.getThreadNum();
//ctx - JMeter Context variable : You get the Thread no. on which your test plan is
log.info("Thread number:"+threadNum);

log.info("Test Plan level User Defined Variable for User ID: "+vars.get("userId"));
vars.put("userId", "bean"+threadNum);
```

```
20-07-06 10:56:18,001 INFO o.a.j.u.BeanShellTestElement: Test Plan level User Defined Variable for User ID: bean
20-07-06 10:56:18,013 INFO o.a.j.t.JMeterThread: Thread started: Thread Group 1-3
20-07-06 10:56:18,016 INFO o.a.j.u.BeanShellTestElement: Print 2
20-07-06 10:56:18,018 INFO o.a.j.u.BeanShellTestElement: Sum: 24
20-07-06 10:56:18,019 INFO o.a.j.u.BeanShellTestElement: After Hi: Hi
20-07-06 10:56:18,019 INFO o.a.j.u.BeanShellTestElement: After Bye: Bye
20-07-06 10:56:18,020 INFO o.a.j.u.BeanShellTestElement: Thread number:2
20-07-06 10:56:18,020 INFO o.a.j.u.BeanShellTestElement: Test Plan level User Defined Variable for User ID: bean
```

Now, new user ID will be passed all time (Thread number will be concatenated with BEAN).

The screenshot shows the JMeter Test Plan tree on the left with a 'User Registration' scenario expanded. Under 'User Registration', there is a 'User account creation' step. This step is expanded to show an 'HTTP Request' configuration. The 'Path' field is set to '/mercurycreate\_account.php'. In the 'Parameters' tab, there is a table with the following data:

Name:	Value
city	Hisar
state	Haryana
postalCode	125001
country	92
email	<b>    \${userId}</b>
password	rajatt95
confirmPassword	rajatt95
register.x	30
register.y	9

The three screenshots below show the 'Result Tree View' for the 'User account creation' step across three consecutive thread groups. Each screenshot shows the 'Request' tab with the following request details:

```

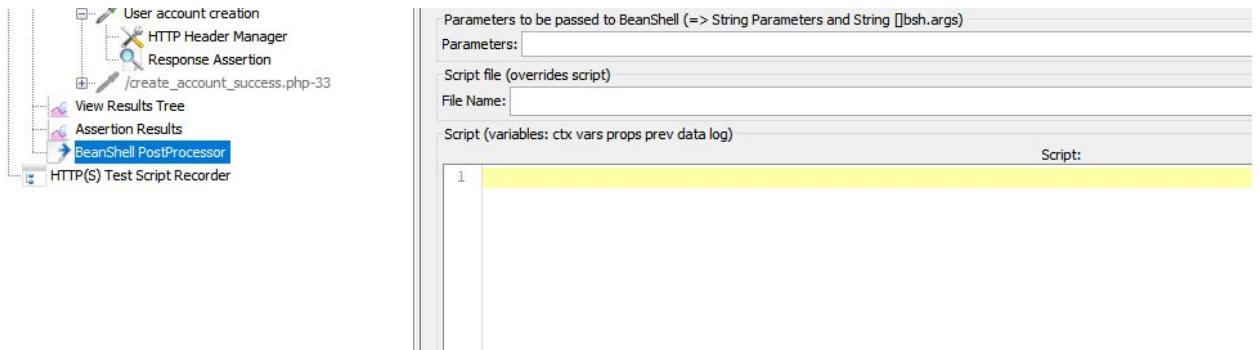
1 POST http://newtours.demoaut.com/mercurycreate_account.php
2
3 POST data:
4 mercury=process&firstName=Rajat&lastName=Verma&phone=9466311644&userName=rajatverma95%40gmail.
com&address1=Sector-15A&address2=&city=Hisar&state=Haryana&postalCode=125001&country=92&email=
bean1&password=rajatt95&confirmPassword=rajatt95&register.x=30&register.y=9
5
6 [no cookies]
7

```

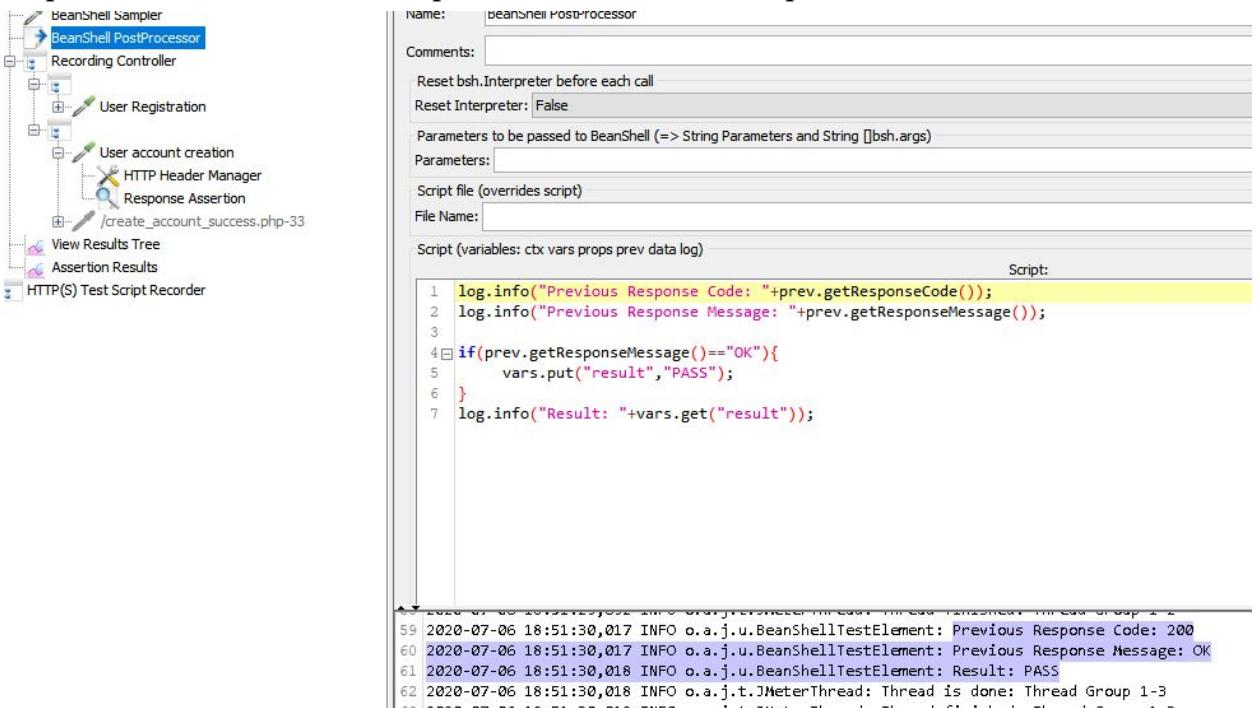
The 'Response Data' tab shows the response body for each iteration, which is identical to the request body.

## BeanShell PostProcessor -

- Right click on Thread Group -> Add -> Post Processors -> BeanShellPostProcessor



- Props - Properties, vars- JMeter variables
- Properties can be defined as Test plan level and Thread Group level as well



=====  
===== Study: Day-10 (July 07, 2020) =====  
=====

===== 16. Integration of Selenium with Jmeter =====

**Eclipse -**

- Maven Project -> SeleniumJmeter
- Pom.xml ->

```
<dependencies>
    <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-server -->
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-server</artifactId>
        <version>3.5.3</version>
    </dependency>
```

- Jmeter supports JUnit
- Write 1 basic Selenium script -> (Open FireFox Browser and get the title of the Page)

The screenshot shows the Eclipse IDE interface. The Project Explorer view on the left displays a Maven project named "Selenium-Jmeter" with its structure: src/main/java, src/test/java, JRE System Library [J2SE-1.5], Maven Dependencies, JUnit 4, src, target, and pom.xml. The Java Editor view in the center contains the code for a JUnit test class:

```
1 package com.learning.Selenium_Jmeter;
2
3 import org.junit.Test;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.firefox.FirefoxDriver;
6
7 public class GetTitle {
8
9     @Test
10    public void testing() {
11
12        System.setProperty("webdriver.gecko.driver",
13            "D:\\Work in Local machine\\Softwares\\geckodriver-v0.26.0-win64\\geckodriver.exe");
14        WebDriver driver = new FirefoxDriver();
15
16        driver.get("https://www.flipkart.com/");
17        System.out.println("Page Title: " + driver.getTitle());
18
19        driver.quit();
20    }
21
22 }
```

The Console view at the bottom shows the execution output:

```
<terminated> GetTitle [JUnit] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (07-Jul-2020, 9:37:41 am)
1594094864936 mozrunner::runner      INFO  Running command: "C:\\Program Files\\Mozilla Firefox\\firefox.exe" "-
JavaScript error: resource://gre/modules/XULStore.jsm, line 66: Error: Can't find profile directory.
1594094916290 Marionette      INFO  Listening on port 58580
Jul 07, 2020 9:38:37 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Page Title: Online Shopping Site for Mobiles, Electronics, Furniture, Grocery, Lifestyle, Books & More. Best Offers!
```

- Export -> Java folder -> JAR file



- D:\Work in Local machine\Softwares\apache-jmeter-5.3\lib\junit -> paste that .JAR file
- Now, I can access this JAR using JMeter UI (Restart the JMeter if this JAR is not reflected in JMeter)
- JMeter -> Add -> Thread Group
- Add -> Sampler -> **JUnit Request**
- If you get any error in creating JUnit request, then, download the Plugin

**Selenium/WebDriver Support**

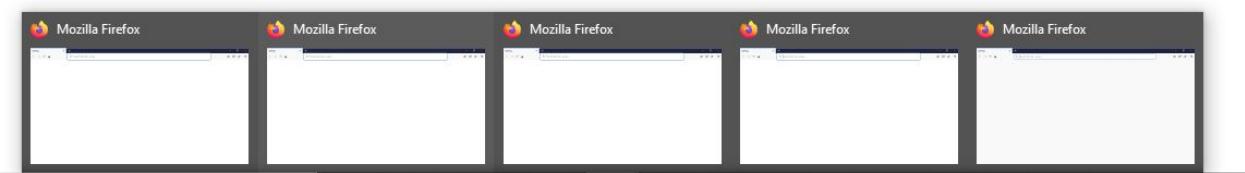
Vendor: *JMeter-Plugins.org*

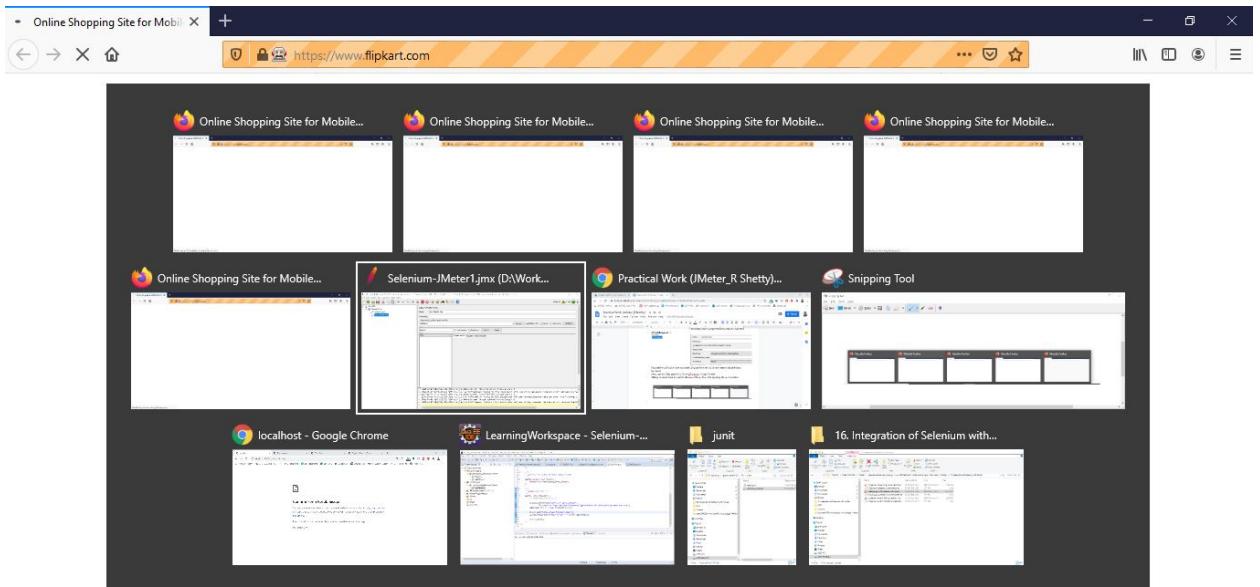
This plugin allows testing real browser behavior using Selenium WebDriver.

**JUnit Request**

Name:	JUnit Request
Comments:	
<input checked="" type="checkbox"/> Search for JUnit 4 annotations (instead of JUnit 3)	
Package Filter:	
Classname:	com.learning.Selenium_Jmeter.GetTitle
Constructor String Label:	
Test Method:	testing

- You need to select the class and method name here (in which you created the Selenium test case)
- Now, run this Test plan for 5 Users (.Options -> Log Viewer)
- Taking so much time to load the Browser (Every time, it is opening the new window)





Performing a TLS handshake to img1a.flixcart.com...

**Text**

- JUnit Request

**Sampler result**   **Request**   **Response data**

Thread Name: Thread Group 1-2  
Sample Start: 2020-07-07 10:22:36 IST  
Load time: 171585  
Connect Time: 0  
Latency: 0  
Size in bytes: 0  
Sent bytes: 0  
Headers size in bytes: 0  
Body size in bytes: 0  
Sample Count: 1  
Error Count: 0  
Data type ("text"|"bin"|""): text  
Response code: 1000  
Response message: Test successful

SampleResult fields:  
ContentType:  
DataEncoding: null

## NOTE:

- It's not recommended to go with Selenium and JMeter scripts
- Because, if you want to run the test with 1000 Users, so 1000 browser windows will open, that will blast your machine and you won't be able to get the exact performance matrix
- System will get hanged (if you open 100 instances at a time)
- But, you can use **Headless Browser** concept, i.e., HTML Unit Driver (work will go without opening the Browser)

## - Eclipse

```
8 WebDriver driver = new HtmlUnitDriver();
9 driver.get("https://www.flipkart.com/");
10 System.out.println("Page Title: " + driver.getTitle());
11
12     driver.quit();
13 }
```

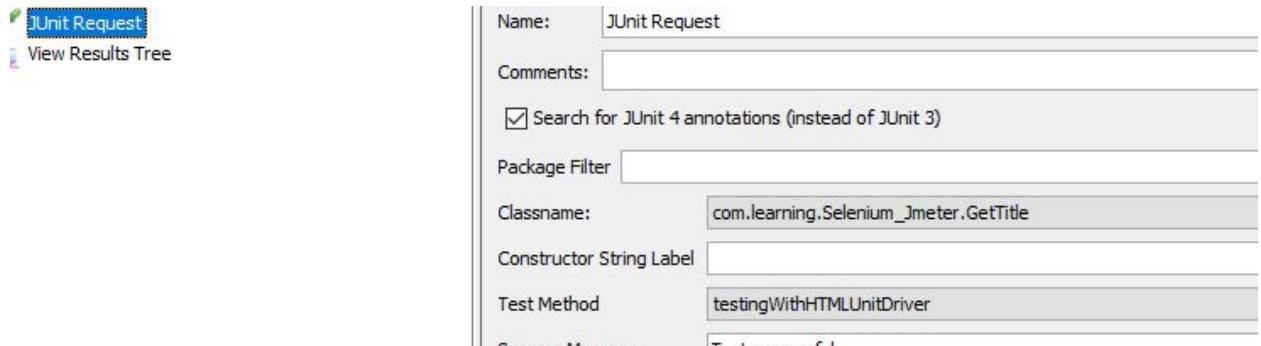
- Run as -> JUnit Test (You must get the Title)

The screenshot shows the Eclipse IDE interface with the Java code for a JUnit test. The code uses the HtmlUnitDriver to navigate to the Flipkart homepage and print the page title. Below the code, the Eclipse toolbars and various perspectives like Markers, Properties, and Data Source Explorer are visible. The Console tab shows the execution output, including a warning about cookie rejection and the correct page title.

```
8
9     @Test
10    public void testingWithHTMLUnitDriver() {
11
12        /*
13         * System.setProperty("webdriver.gecko.driver",
14         * "D:\\Work in Local machine\\Softwares\\geckodriver-v0.26.0-win64\\geckodriver.exe"
15         * ); WebDriver driver = new FirefoxDriver();
16         */
17
18        WebDriver driver = new HtmlUnitDriver();
19        driver.get("https://www.flipkart.com/");
20        System.out.println("Page Title: " + driver.getTitle());
21
22        driver.quit();
23    }
24
25 }
```

Markers Properties Servers Data Source Explorer Snippets Console JUnit  
<terminated> GetTitle [JUnit] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (07-Jul-2020, 10:30:58 am)  
Jul 07, 2020 10:32:00 AM org.apache.http.client.protocol.ResponseProcessCookies processCookies  
WARNING: Cookie rejected [T="T1159409811806900198430746177853482600773867047549113354555742113469", version:0, domain:.flipkart.net,  
Page Title: Online Shopping Site for Mobiles, Electronics, Furniture, Grocery, Lifestyle, Books & More. Best Offers!

- Then, repeat the procedure (Export a JAR file of the Project and put it in .....lib/junit folder)



## =====17. REST API Load Testing using Jmeter=====

### REST -

- Representational State Transfer
- Lightweight alternative to SOAP

### SOAP -

- SOAP is a Protocol for web services communication, REST is an architecture.
- Protocol means a **set of standards** to be followed while sending the messages.

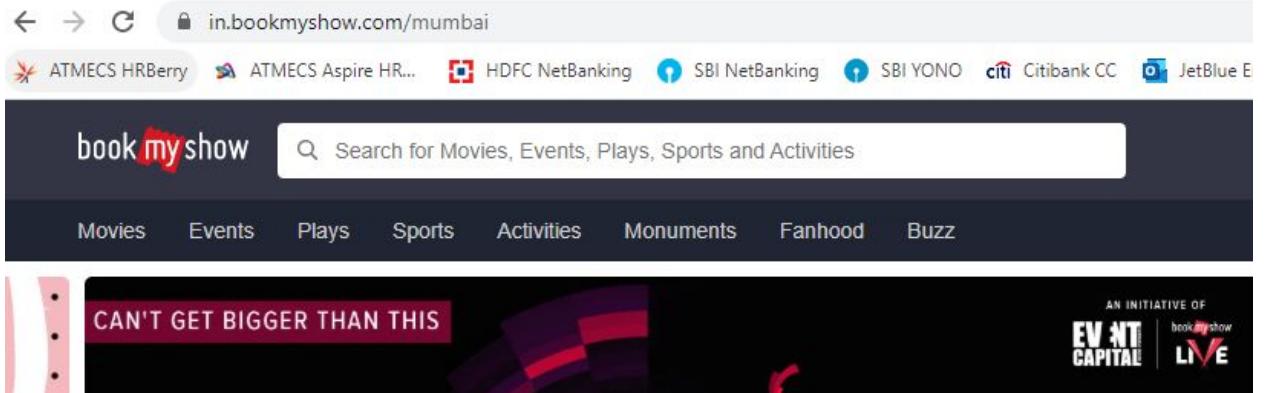
```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
    <soap:Header/>
    <soap:Body>
        <typ:getEmployeeDetails>
            <!--Optional:-->
            <typ:name>${DataGen#Name}</typ:name>
        </typ:getEmployeeDetails>
    </soap:Body>
</soap:Envelope>
```

- SOAP is not the only one Protocol to have communication with Web services;  
WS-\* extensions such as WS-Security, WS-Addressingm WS-ReliableMessaging are associated with SOAP messaging.

### Why REST?

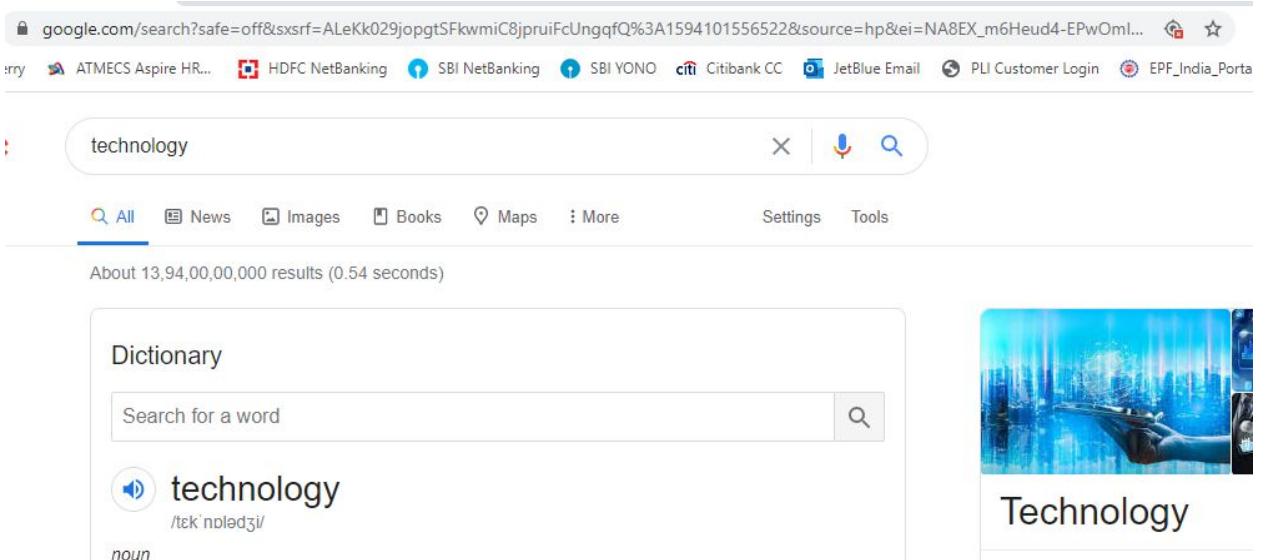
- SOAP becomes very heavy (even when if we want to send 1 data set; then also, we need to send Header, Body, Envelope inn XML format)
- Has less performance degradation

- **REST -**
  - Uses standard HTTP methods to access the resources/operations
  - GET, POST, DELETE, PUT
  - Follows Stateless communication
  - To access the resources, Resource Identifiers are used with the help of different HTTP methods.
  - Resources : Movies, Events, Sports, Activities, and, so on.



### **Steps -**

- You do search in Google
  - For example:



[GBYgB9DmSAQowLjQuNy4wLjJuNi4xmAEAoAEBqgEHZ3dzLXdperABCg&sclient=psy-ab](#)

- No. of parameters are attached in the Request

### HTTP methods -

- GET - Read
- POST - Create
  - Logging facebook...Session is created
  - Resource Identifier will not be passed from the URL
  - Fill credentials and submit the form -> You won't see any change in the URL after successful Login

### Google has exposed many services -

- For Practice, we will use -> <https://developers.google.com/maps/documentation>
- You will get the search Responses like this

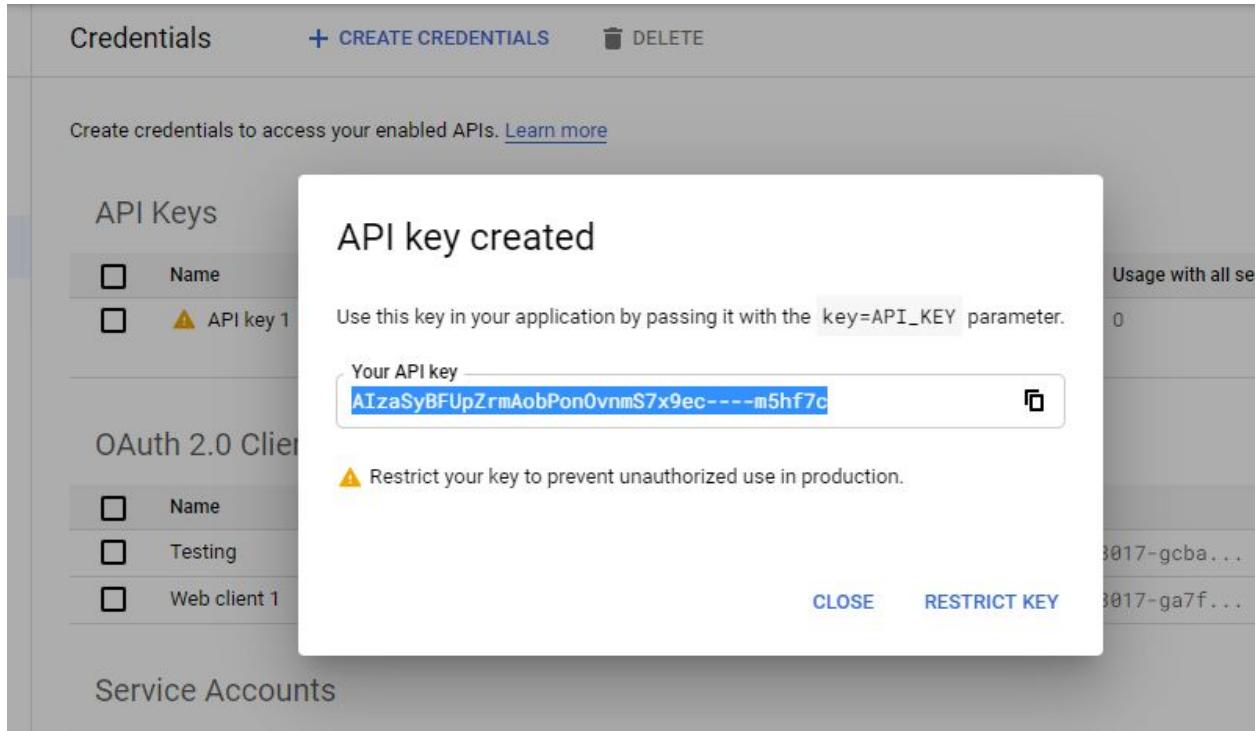
The screenshot shows a browser window displaying a JSON response from the Google Places API. The URL in the address bar is <https://developers.google.com/places/web-service/search#PlaceSearchResponses>. The response object contains a 'candidates' array with one item. The item includes a 'formatted\_address' (140 George St, The Rocks NSW 2000, Australia), a 'geometry' object with 'location' (lat: -33.8599358, lng: 151.2090295) and 'viewport' (northeast: lat: -33.85824767010727, lng: 151.2102470798928; southwest: lat: -33.86094732989272, lng: 151.2075474201073), and a 'name' (Museum of Contemporary Art Australia). The JSON response is displayed in the main content area, with a sidebar containing navigation links for the Google Maps Platform documentation.

### Let's put Load on Google API using JMeter-

- <https://maps.googleapis.com/maps/api/place/nearbysearch/output?parameters>
- <https://maps.googleapis.com> - Address (Server IP)
- <maps/api/place/nearbysearch/output?parameters> - Path to access particular resource

### To Create Google API key -

- <https://cloud.google.com/docs/authentication/api-keys>
- Click on APIs & Services → Credentials → <https://console.cloud.google.com/apis/credentials>
- Create Credentials → API Key



## JMeter -

- Add -> Sampler -> HTTP Request
- Fill all the required values in different fields

**HTTP Request**

Name: REST API Load Test

Comments:

Basic Advanced

Web Server

Protocol [http]: https Server Name or IP: maps.googleapis.com

HTTP Request

GET Path: maps/api/place/nearbysearch/json?

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipart/form-data  Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type
location	42.3675294,-71.186966	<input type="checkbox"/>	text/plain
radius	10000	<input type="checkbox"/>	text/plain
key	AIzaSyBFUpZrmAobPonOvnms7x9ec----m5hf7c	<input type="checkbox"/>	text/plain

**HTTP Request**

Name: REST API Load Test

Comments:

Basic Advanced

Web Server

Protocol [http]: https Server Name or IP: jsonplaceholder.typicode.com/ Port Number:

HTTP Request

GET Path: users Content encoding:

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipart/form-data  Browser-compatible headers

Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type

The screenshot shows two JMeter result windows side-by-side. The top window displays the 'Response Body' tab, showing a JSON response structure. The bottom window displays the 'Sampler result' tab, showing various performance metrics.

**Response Body:**

```
[  
 {  
 "id": 1,  
 "name": "Leanne Graham",  
 "username": "Bret",  
 "email": "Sincere@april.biz",  
 "address": {  
 "street": "Kulas Light",  
 "suite": "Apt. 556",  
 "city": "Gwenborough",  
 "zipcode": "92998-3874",  
 "geo": {  
 "lat": "-37.3159",  
 "lng": "81.1496"  
 },  
 "phone": "1-770-736-8031 x56442",  
 "website": "hildegard.org",  
 "company": {  
 "name": "Romaguera-Crona",  
 "catchPhrase": "Multi-layered client-server neural-net",  
 "bs": "harness real-time e-markets"  
 }  
 },  
 {
```

**Sampler result:**

- Thread Name: Thread Group 1-4
- Sample Start: 2020-07-07 16:46:02 IST
- Load time: 497
- Connect Time: 341
- Latency: 490
- Size in bytes: 6530
- Sent bytes: 138
- Headers size in bytes: 872
- Body size in bytes: 5658
- Sample Count: 1
- Error Count: 0
- Data type ("text"|"bin"|""): text
- Response code: 200
- Response message: OK

HTTPSSampleResult fields:  
ContentType: application/json; charset=utf-8  
DataEncoding: utf-8

Performance is good when Load Time decreases from top to bottom.

=====

===== Study: Day-11 (July 08, 2020) =====

=====

===== 18. Monitoring Server performance =====

Browser -> Google.com -> Server is there to handle all the requests

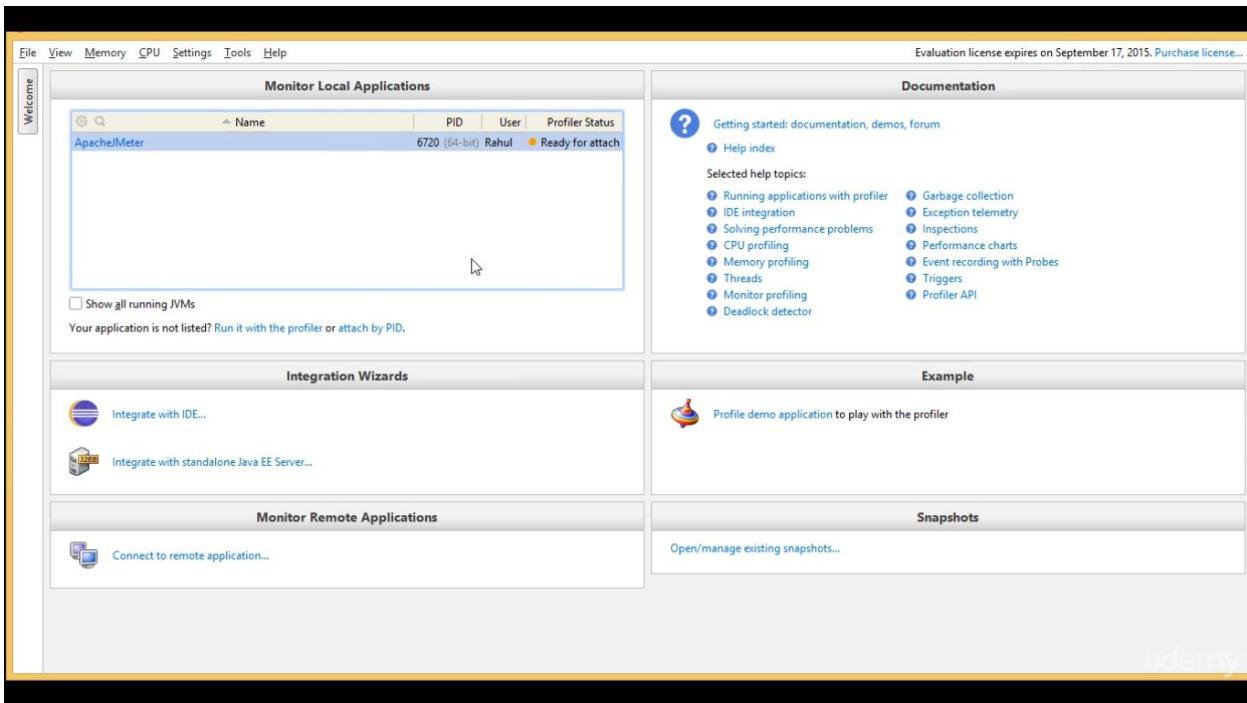
Memory consumption, CPU working

Till now, we were observing the Host server performance,  
But, **Client server** is also important (the machine from where, the services are being hit).

We are going to do CLIENT SERVER monitoring.

For Monitoring the Server Parameters, we'll use the Tool: YourKit Java Profiler (<https://www.yourkit.com/java/profiler/>).

Download and install it and go to its bin folder and run Profiler.exe.



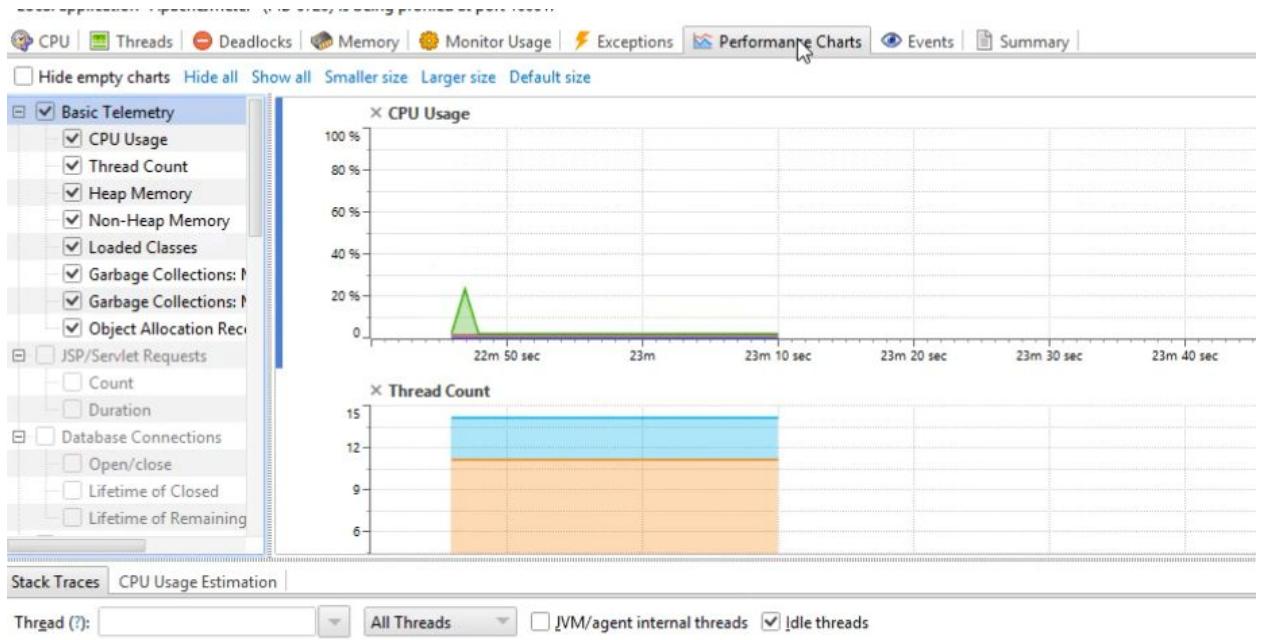
- It will monitor all the applications running on JVM.
- Apache JMeter is mentioned in the list because it is already running in my machine.
- IDE Eclipse also runs on JVM, if Eclipse is also open in local machine, then, Eclipse will also be listed in the List
- Resources consumption

## Steps -

- Right click on Ready to Attach -> Attach to Application



- After attachment,



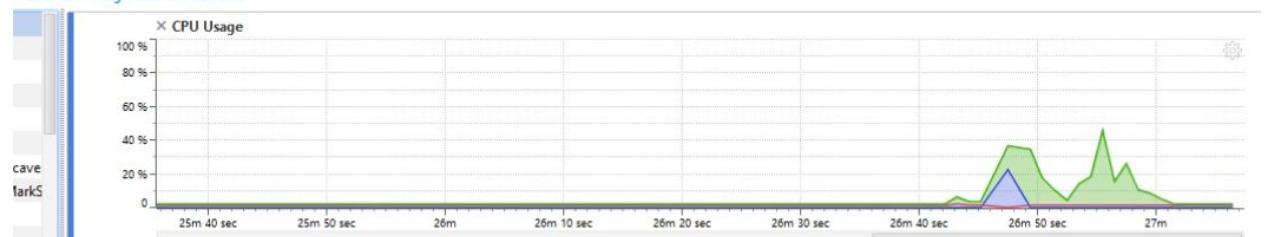
- When you trigger some script in JMeter, you can observe that CPU consumption has gone to 40% at 1 time
- (40 % is ok, but, whole CPU consumption should not go for any single tool)
- This kind of tool is for doing observations on our system resources consumption
- You should not only keep an eye on server side of the Application, Client side is also very important

) is being profiled at port 10001.

⚠ Some profiling capabilities are not available in attach mode

Memory | Monitor Usage | Exceptions | **Performance Charts** | Events | Summary |

Smaller size Larger size Default size



## =====19. Bonus lecture Bugzilla - Testing tool for Defect Tracking=====

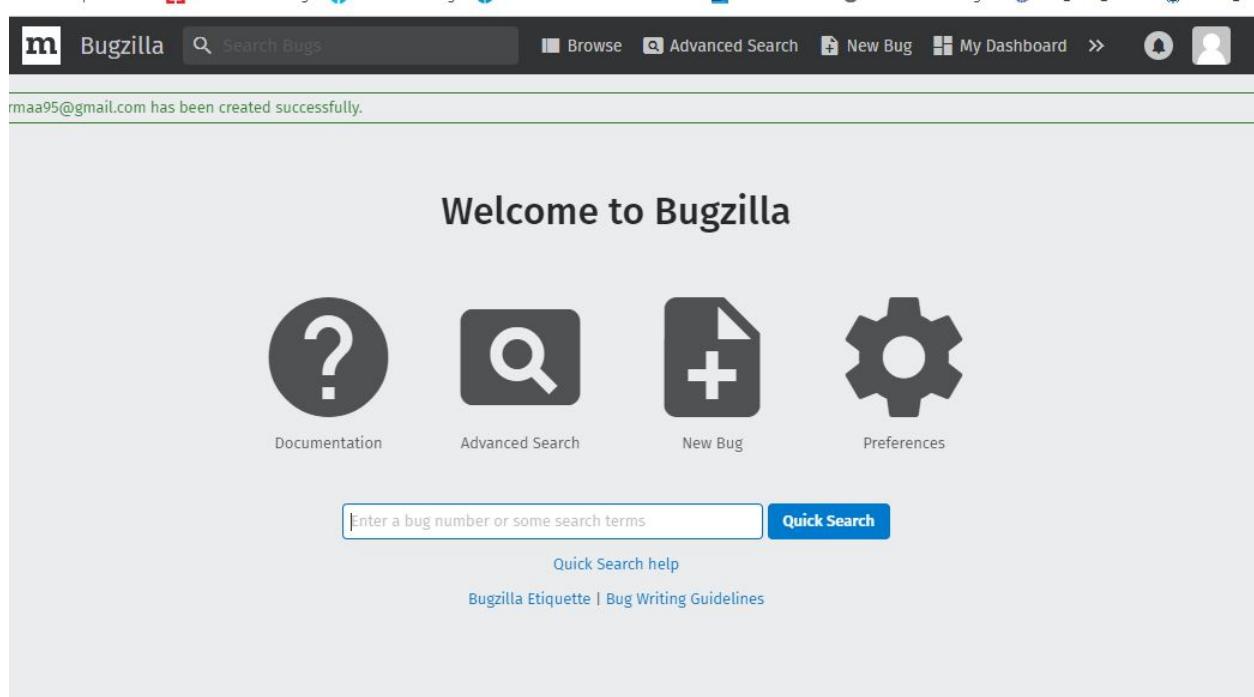
### Bugzilla -

- Bugzilla is a web-based general-purpose **bug tracking system** and testing tool.
- Open-Source
- You can integrate it with your automation testing suites
- Clients may or may not ask you to install it in your local machine.
- Mostly, it is installed in the server side and managed by the Administrative Team (Admin Work)

How does Bugzilla work as a Bug management system?

- Bugzilla Team has deployed Demo link (they have hosted on their web server)
- You don't need to install it in your local machine

Create new account (<https://bugzilla.mozilla.org/createaccount.cgi>)



## File a Bug -> Logging a defect -

 I need technical support  
For technical support or help getting your site to work with Mozilla.

 Report an issue with a website that I use

**None of the above; my bug is in:**

 <b>Firefox</b> For bugs in Firefox, the Mozilla Foundation's web browser. ( <a href="#">more info</a> )	 <b>Firefox for Android</b> Firefox for Android is a mobile version of Firefox for Android devices. ( <a href="#">more info</a> )	 <b>Firefox for iOS</b> Firefox for iOS is the Firefox mobile experience developed for the iOS platform.
 <b>Thunderbird</b> Email client originally developed by the Mozilla Foundation and now maintained by its community. ( <a href="#">more info</a> )	 <b>Other Products</b> Other Mozilla products which aren't listed here	

- Under which category, your bug came?

## Enter A Bug

- Please fill out this form clearly, precisely and in as much detail as you can manage.
- Please report only a single problem at a time.
- [These guidelines](#) explain how to write effective bug reports.

<b>Summary:</b>	This is a test for Logging the Defect/Bug.	
<b>Product:</b>	Firefox for Android ( <a href="#">Change</a> )	<b>Version:</b> unspecified 
<b>Component:</b> (List)	General	
For bugs in Fennec which do not fit into other more specific Fennec components.		
<b>What did you do? (steps to reproduce)</b>		
<b>What happened? (actual results)</b>		

- You need to give the details about the Bug/issue?
- After you submit all the details

The screenshot shows a Bugzilla bug detail page for bug 1651264. At the top, there are buttons for 'Copy Summary', 'Follow', 'View ▾', 'New/Clone ▾', and 'Edit Bug'. The bug status is 'Open' and it was 'Opened just now'. The title of the bug is 'This is a test for Logging the Defect/Bug.' Below the title, there are sections for 'Categories' and 'Tracking'. Under 'Categories', the product is 'Firefox for Android', component is 'General', version is 'Firefox 71', type is 'defect', priority is 'Not set', and severity is '--'. Under 'Tracking', the status is 'UNCONFIRMED'. There are also sections for 'People' (reporter: rajatvermaa95, unassigned), 'Details', and 'Security' (Security-Sensitive Mobile Client Bug). A 'Attachments' section with a 'Attach New File' button is also present. At the bottom right, there are links for 'Add Comment ▾' and 'Timeline ▾'.

- Open Bug 1651264 means -> Status: Open, Bug no. 1651264
- All necessary information is listed there.
- You can edit/update the Bug (created by you) anytime

#### **NOTE -**

These notes doesn't cover last 2 folder contents:

- 19. Bonus lecture Bugzilla - Testing tool for Defect Tracking
- 20. Jmeter older version topics(Obsolete)

**That's all for JMeter.**