Jmeter

1. Installation

* Download jdk
* Download jmeter using- “Apache Jmeter”
* Download jmeter plugin- 3 basic graph, 5 additional graph
* From 3 basic graph copy jar file and paste in jmeter lib folder and copy ext file from plugin to jmeter
* From bin folder open jmeter executable file

1. Create thread

* Right click on test plan 🡪 Add🡪 thread🡪 Thread group
* Add number threads (users ) and loop count

1. Add sampler

* Right click thread group 🡪 Add🡪 sampler🡪 Http

1. Add listener

* Right click on test plan 🡪 Add🡪 listener🡪 view result tree

1. Add timer

* Right click on test plan 🡪 Add🡪 timer🡪 Gausian random timer

1. Assertion in Jmeter

* Right click on test plan 🡪 Add🡪 Assertion🡪 Response Assertion
* Add response and status code

1. Config element- It is manage header, cache and cookies

Jmeter Performance testing :

* Performance testing : Ensure software application will perform well under a particular load.
* Advantages of Jmeter

1. Open source
2. Cross platform support
3. Script is not essential
4. GUI is more user friendly
5. Work on any OS

* Install Jmeter:-

1. Download from “download jmeter” and download .zip file from binarary
2. Download jdk as per jemeter required

* Record and playback

1. Right and click on test plan
2. Click on add
3. Select non test element
4. Select http test script recorder
5. Test plan creation- select target controller is ‘http test script recorder
6. Request filtering – Add suggested excluded select
7. Configure firefox – foto setting – select manual seeting 🡪 give port number and local page address
8. Start case and create certificate and provide to firefox browser
9. Start the case and record by start noemal flow of our application

* Chrome jmeter config.

1. Download blazemeter extension - <https://chrome.google.com/webstore/detail/blazemeter-the-load-testi/mbopgmdnpcbohhpnfglgohlbhfongabi?hl=en>
2. Import file from :- file 🡪 open 🡪 choose file and open

* Thread Group In Jmeter

1. Able to load using thread group
2. Thread properties
3. Total number threads : total user
4. Ramp up period : with time user on server ex total user 50 and ramp up period 5 then 10 user 1sec, 10 user 2sec…
5. Loop count : total count to run
6. Infinite : run case infinite
7. Duration: how much time test will run

* Listener in Jmeter : to generate report

1. Add listener view result tree
2. Add aggregate report
3. Add aggregate graph

* Jmeter load parameters

1. Sample : sample heat in total schedule time
2. Average: total average time required to complete all request
3. Min : Minimum time required to request
4. Max: max time to get response
5. Error % : failed %
6. Throughput : no of request processed/hr
7. 90 % line : 90 % request time
8. 95% line : 95 % request time
9. 99% line : 99 % request time
10. Median : max-min value
11. Std deviation : fluactuation from avg. response time. Std deviation is less than half of average.

* Additional plugins to add real time load

1. Download plugins manager : <https://jmeter-plugins.org/wiki/PluginsManager/>
2. And paste file in lib/ext folder
3. Go to option 🡪 select custome thread group and lock file config select and click on apply changes
4. Restart the jmeter

* Concurrence thread group

1. Ramp up step count – means it complete request as enter ex 3 then 3 time it completed
2. Ramp up time: total time schedule for request
3. Target concurrency : total user
4. Told target rate time : total time user hold on server

* Ultimate thread group

1. Start thread count: total user
2. Initial start up time : no user start heat
3. Startup time : start the request heat time
4. Hold time : hold all request in resp. time
5. Ramp down time :request close time

* Add cookies manager in Test

1. Cookie store the authentication and its work like browser in jmeter
2. If cookies not added then case correct but incorrect response
3. Add HTTP cookie manager : right click on thread group 🡪 go to “config element” 🡪 select HTTP cookie manager

* Assertion In Jmeter

1. To validate the response (text/status code/header)
2. Right click on test and add 🡪 Assertion 🡪 HTTP assertion
3. Http Assertion handle 🡪 response, header, contains
4. Size Assertion : check size in bytes of response. 🡪 right click on thread group 🡪 assertion 🡪 Size assertion
5. Add assertion result from 🡪 listener 🡪 Assertion result
6. Duration Assertion -find out maximum benchmark of request . 🡪 right click on thread group 🡪 assertion 🡪 duration assertion

* **Controller in JMETER**

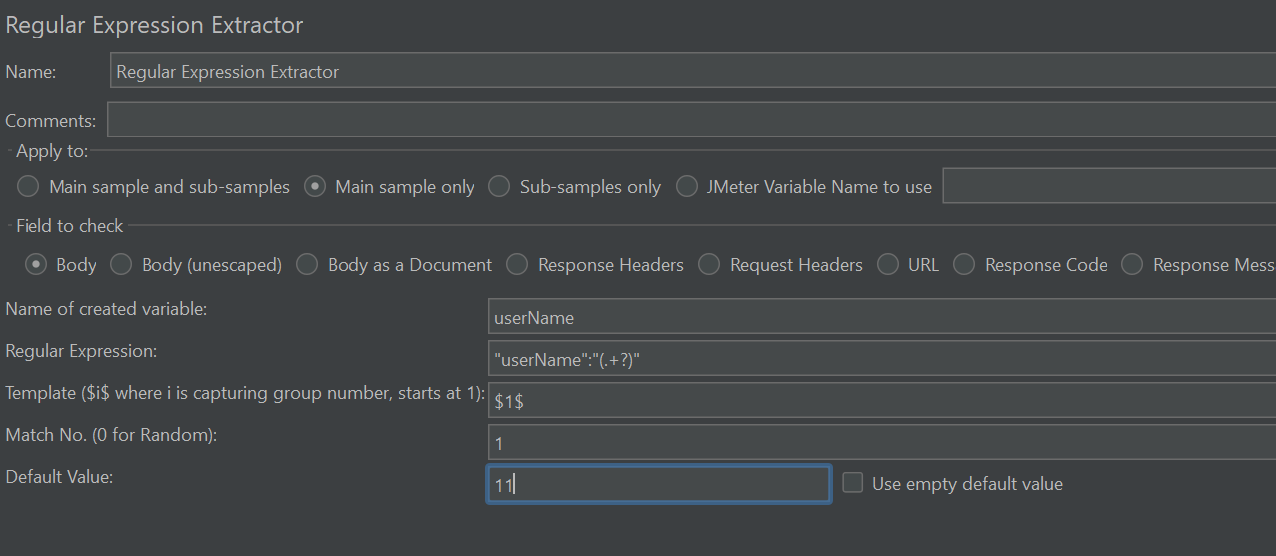
1. Collection of sampler/request. It store all request
2. ***Controller*** :
3. ***Recording controller*** : record all the request and store
4. ***Transaction controller*** : Store all request. Show result module wise/ request wise
5. ***Simple controller :*** simply store request and show report
6. Module controller : Used for reuse request
7. ***Interleave controller*** : User for run the request top to bottom (every time pick up one in iteration)
8. ***Runtime controller*** : run the request provide time continuously
9. ***Random Controller:*** pick any request and run
10. ***If Controller***: if request condition ok then run request only.
11. We declare variable in test plan user defined variable and validate from if controller ex “${store}”==”12”
12. ***Loop Controller***: run request as per enter value.
13. ***While controller***: if condition is true it is run till condition not false

* **Timer’s in JMETER**

1. ***Constant Timer*** : wait to execute next loop for defined time. If time declare 300 milli sec then thread time increase also same.
2. ***Gaussian Random Timer*** : Constant timer + deviation in second random (Ex= constant 200+ dev. 100) Note : Add result table from listener to view table.
3. Constant throughput Timer : calculate maximum request server in min

* **Regular expression in JMETER** – Useful for dynamic data

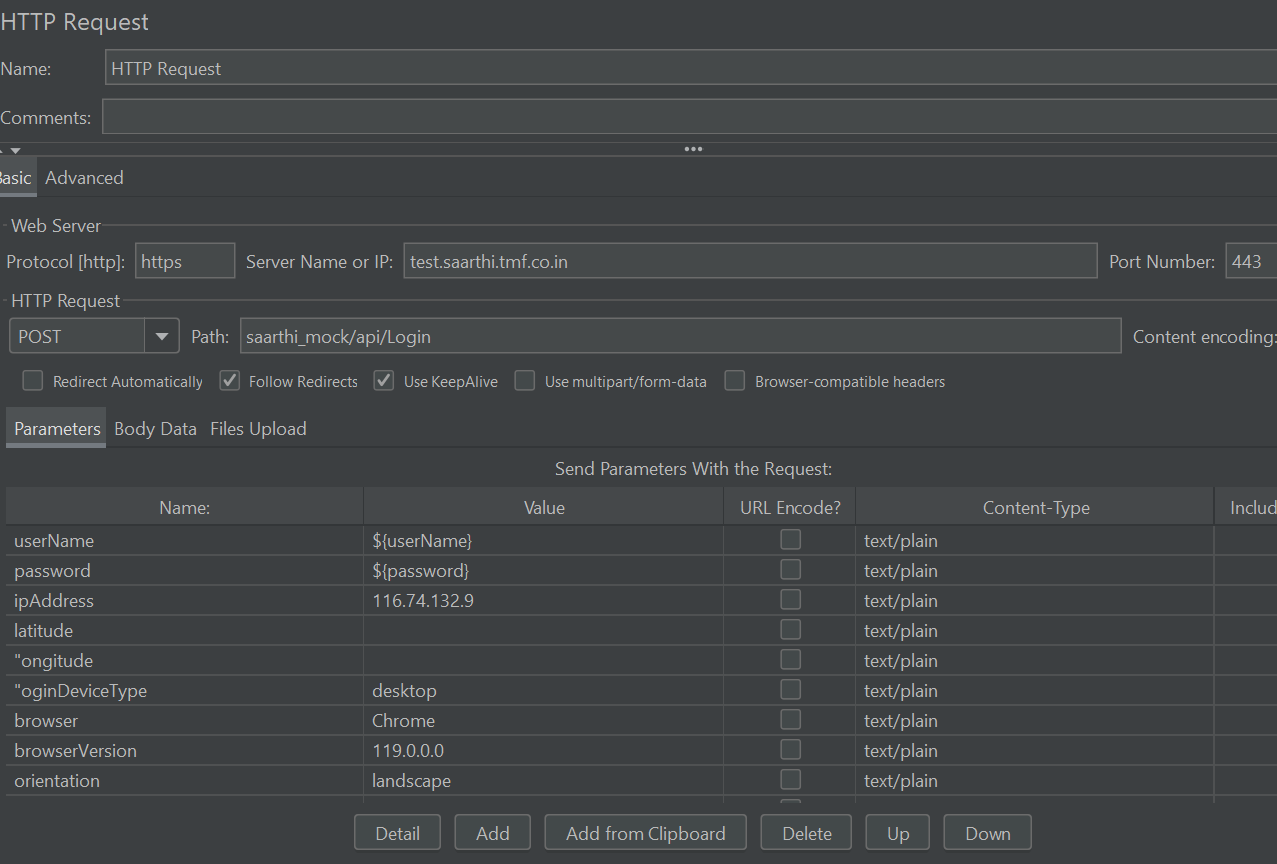
1. Add test plan, Add request using not test element, Add cookies manager, add view result tree listener 🡪 add regular expression extractor from post processor
2. Add variable in reg, expression extractor



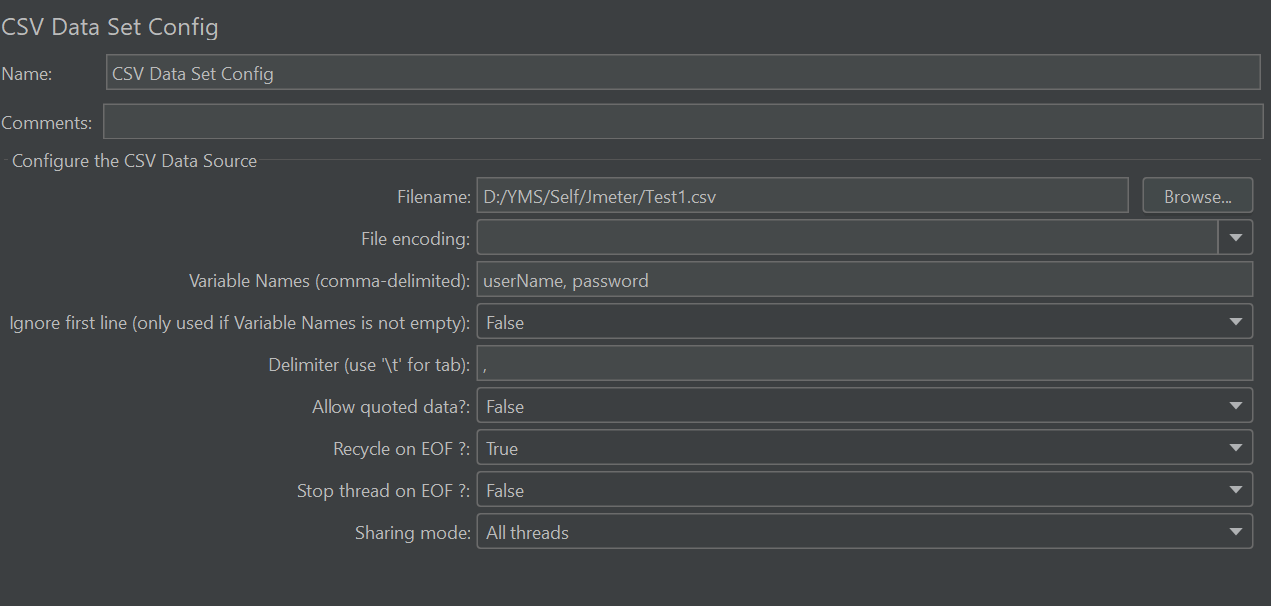
1. Debug Sampler : use for what value is pass is showing.

* **Data driven in JMETER**

1. Create one Test plan and add thread
2. Add one HTTP request from thread 🡪 sampler 🡪 HTTP request
3. Add information ex.



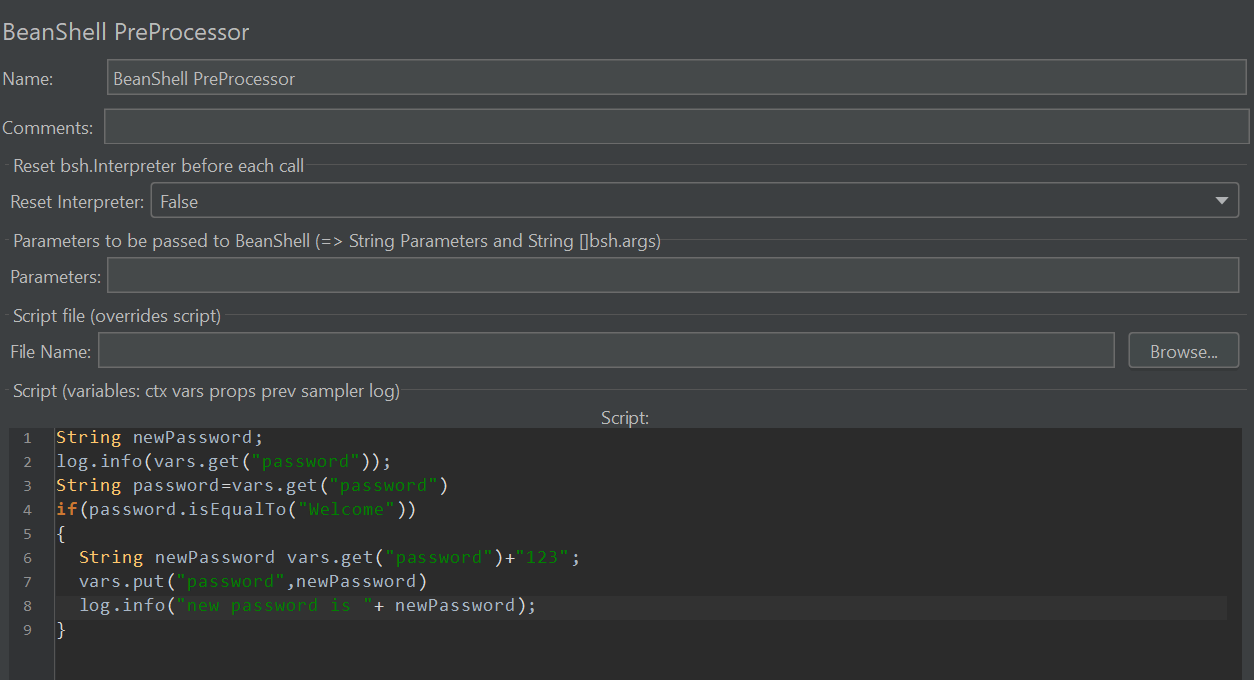
1. Create one csv excel and add data
2. Create CSV data set config from and upload excel and upload data ex.



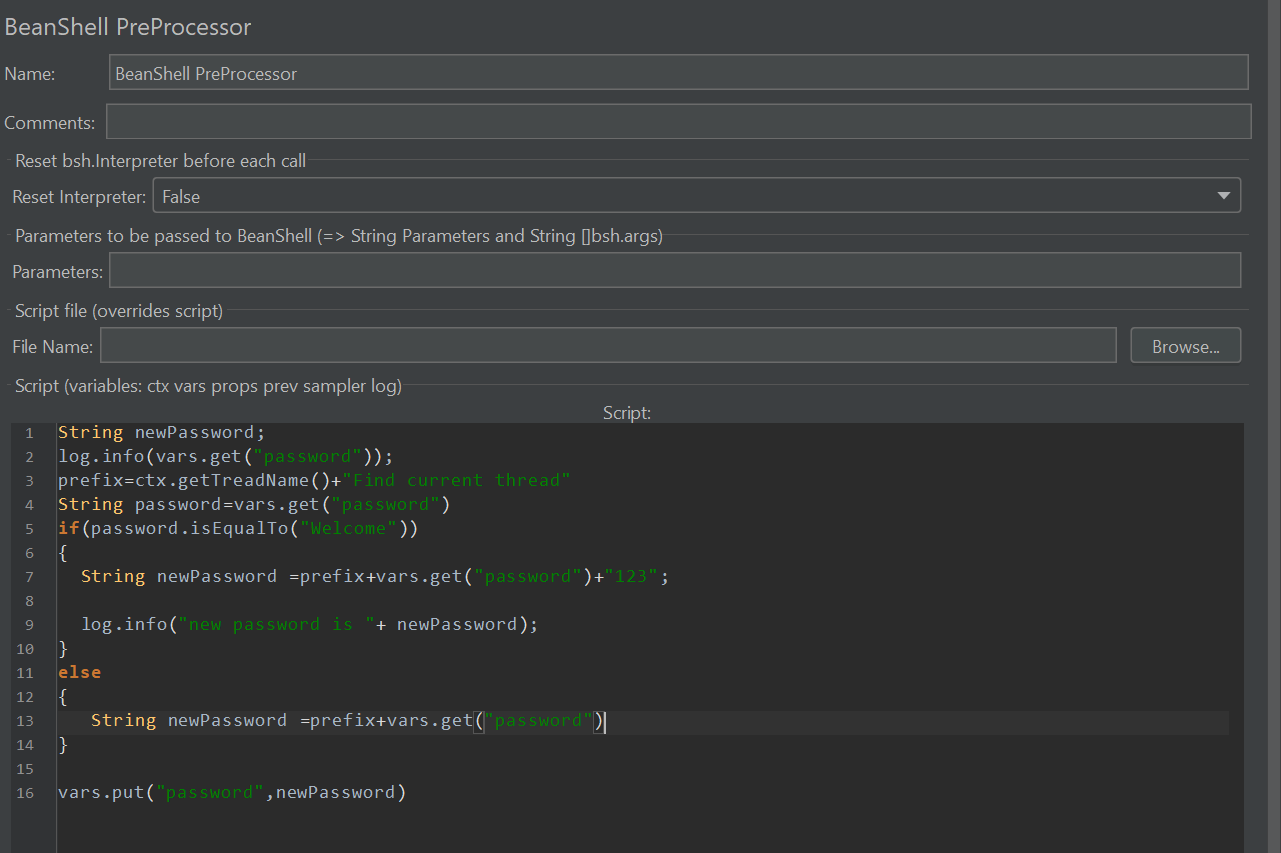
1. Add view tree result and assertion for check result

* **Bean Shell pre processor in JMETER**:

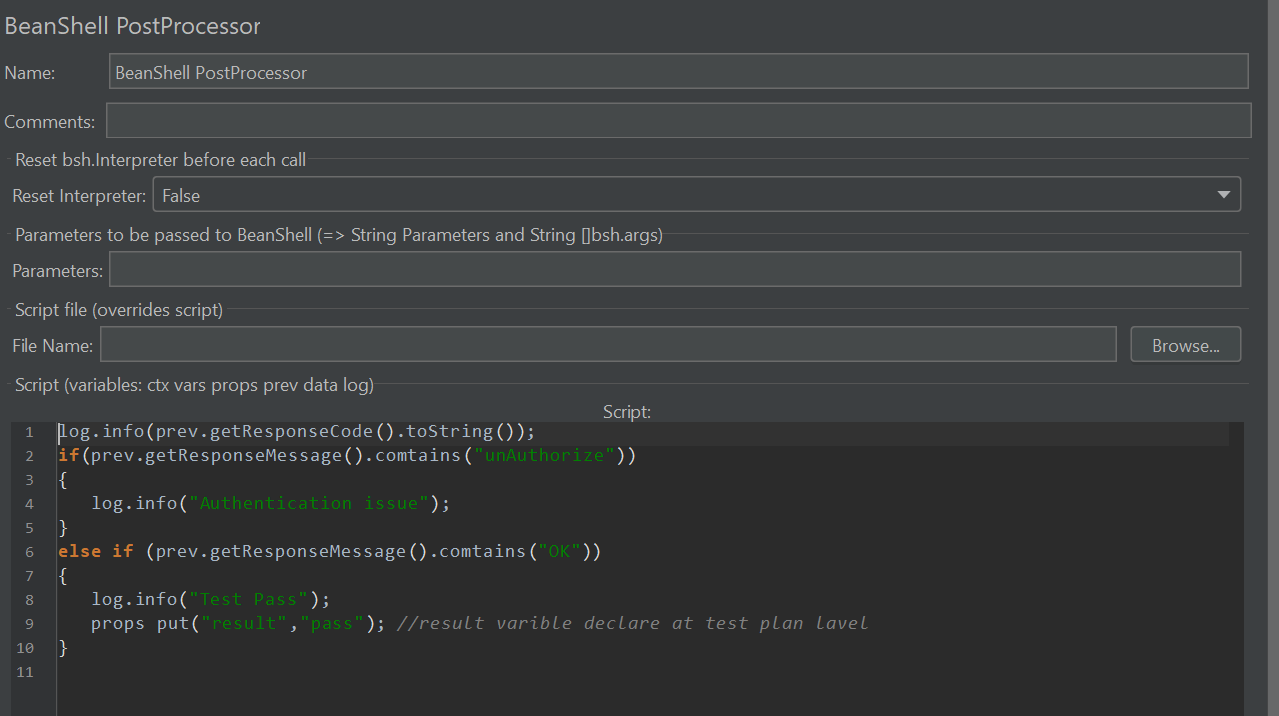
1. After same above process add bean shell from pre process
2. It is java language to generate dynamic data with validation
3. EX.



1. Bean shell methods:
2. Ctx.getTreadNumber //find current thread and generate unique data
3. Ex.



1. Validate response using bean shell postprocessor ex.



* Slave in JMETER

1. First opne jmeter.prperties file and true : server.rmi.ssl.disable=true
2. Set Server in remot\_Host
3. Restart Jmeter
4. Select from run – start server

* Run Jmeter test using Non GUI (Command prompt)

1. Open command prompt as run administrator
2. Provide path using cd
3. Run the test plan using : jmeter -n -t filename.jmx -l c:\results.jtl …..(t means test plan, -I used for result)