JMeter

**JMeter Elements:**

Thread Group, Samples, Listeners, and configuration.

**Thread Group** – Collection of threads. Each thread represents one user using the AUT. It simulates one user to the server.

**Samplers** – different types of requests send by Thread group. The user request could be FTP request, HTTP request, JDBC request etc.

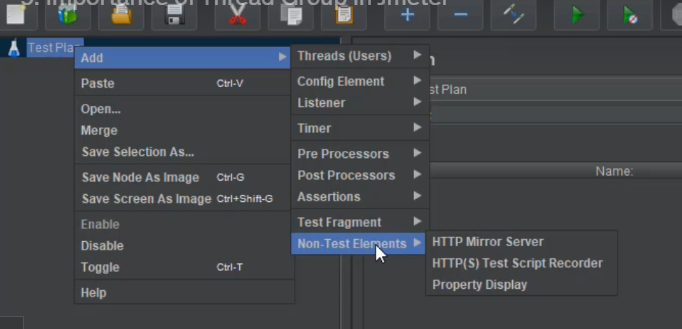
**Listeners** – Shows the result of your test execution. Show in diff format like log file, graph (servers response time on a graph), Tree view or table view.

**Configuration Element** – used for setting the default variable which are used in samplers like config login element, HTTP request default and FTP request default.

**Test plan** – It consists of all actions and components that you need to execute your performance script. It allows user to create only one project at a time and for different project user must create a new test plan.

**How to create a Thread Group:** Right click on test plan 🡪 Add 🡪 Thread(users) 🡪 Thread Group.

**How to record in JMeter:**



**Aggregate report under Listeners:**

A screenshot of a computer

Description automatically generated

Samples – No of users hit the specific request.

A screenshot of a computer screen

Description automatically generated

**Additional plugins for simulating real time load:**

Download jars from <https://jmeter-plugins.org/wiki/PluginsManager/>

**Concurrency ThreadGroup – almost similar to normal thread group, Ultimate ThreadGroup 🡪 When you want to ramp down the users at a specific interval. Imp thread group under custom thread group.**

If user adds one after the other plugin by downloading respective jars then it will consume a lot of time instead user can download the “plugin manager” jar from the above path and place the downloaded jar in apache/lib/ext path. Then if user restarts the JMeter, Options 🡪 Plugins Manager.

A screenshot of a computer

Description automatically generated

Navigate to Available Plugins tab, search for respective plugin let us say “Custom Thread Group”, check it and click apply changes and restart JMeter.

A screenshot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

**How to add a test plan:**

Add a test plan 🡪 Add a thread group 🡪 add the recorded steps to the thread group 🡪 add a listener 🡪 run and view the result using the listeners.

**HTTP cookie manager:**

Right click on test plan 🡪 Add 🡪 Config Element 🡪 HTTP Cookie Manager.

**How to add a assertion:**

Right click on the specific request 🡪 Add 🡪 Assertions 🡪 Response Assertion 🡪 select either text response (add the text you want to verify) or response code (add the response code you want to verify)

**How to see the assertion result:**

Right click on test plan 🡪 Add 🡪 Assertions 🡪 Assertion Result.

**Controller:**

Set of similar request / samplers are grouped under one folder called controller. It is useful when your requirement is to find the total time taken for one set of action. Let us say a flow with steps like **login** 🡪 **some actions on home screen** 🡪 **logout**.

In this case we have 3 controllers i.e., one for each action. So, when you segregate them into separate controller then it is easy to find the total time take for each action.

**How to add controller in test plan:**

Right click on Test plan 🡪 Add 🡪 Threads 🡪 Thread Groups.

Right click on Thread Groups 🡪 Add 🡪 Logic Controller 🡪 Select the desire controller.

**How to record steps in jMeter:**

1. Right click on Test plan 🡪 Add 🡪 Non-Test Elements 🡪 HTTP(s) Test Script Recorder.
2. Right click on Test plan 🡪 Add 🡪 Threads 🡪 Thread Groups.
3. Select test script recorder under test plan 🡪 Request Filtering 🡪 click “Add suggested Excludes”.
4. Click “Test Plan Creation” tab 🡪 Set Target controller = Use Recording Controller”.
5. Right click on Thread Group under test plan 🡪 Logic controller 🡪 Recording controller.
6. Select HTTP(s) Test Script Recorder under test plan 🡪 click start 🡪 move the browser where we have set the proxy port number to 8888.
7. Do the steps that user wants to record.
8. The steps that user performs in web browsers will get recorded under “Recording Controller”.
9. If user wants to segregate the steps, then after login switch to JMeter 🡪 cut all the samplers that are generated under “Recording Controller” and paste it to 🡪 Right click on Thread groups 🡪 Add 🡪 Logic Controller 🡪 Transaction Controller. Paste it on the newly created transaction controller and rename it to Login for user understanding.
10. Similarly, user can segregate the steps for other modules.

**How to get the time taken for one module:**

1. Add Listeners to the above steps under thread groups like Aggregate report, View Results Tree.
2. Simply run the program with thread group is set to 1. The result will be grouped as show below.

A screenshot of a computer

Description automatically generated

1. User can get the total time taken for each module as shown above like Login, Flight search etc.
2. User can also get only the module level report (just login, flight search) instead of sampler level reports (each step under login and flight search module). To get this select the module and check “Generate Parent sample”.
3. A screenshot of a computer

   Description automatically generated
4. Now if user runs the test plan again, then the report will look something like below.

A screenshot of a computer

Description automatically generated

**Difference between Simple Controller and Transaction Controller:**

Both are used to segregate the steps, but transaction level controller gives report for both steps and module level whereas simple controller will give only report at steps level i.e., you can perform step no 4 in the above.

**Module controller:**

When user wanted to rerun the already recorder steps or already written script, then module controller helps to achieve it.

**How to add module controller:**

Right click on Thread group 🡪 Add 🡪 Logic Controller 🡪 Module Controller.

A screenshot of a computer

Description automatically generated

**How to use it to run the existing modules:**

When user clicks the newly added module controller, it will display the test plan and thread group of our current project. Select the desired module and save.

**How to identify whether the module controller worked:**

Simple run the project and user can observe that the aggregate report shows 2 samples under login module as shown below.

A screenshot of a computer

Description automatically generated

**Interleave controller:**

This will run only one step at a time in a loop i.e., if user has 3 steps inside the interleave controller, then on the first loop, it will run only the first step and then in the 2nd loop it will run the 2nd step. It will execute from top to bottom order.

A screenshot of a computer

Description automatically generated

In the above example, under interleave controller, we have added flight search transaction controller 2 time (1 time it will search for London to delhi and 2nd time it will search for America to Delhi). When user runs all modules runs 2 times, but flight search and flight search-2 run only one time.

**Timers:**

The purpose is to give delay between each loop.

**Types:**

Constant Timer 🡪 give a same amount of delay before every loop starts.

Gaussian Random Timer: gives a different amount of delay every time because of the field deviation and it would vary from 0 to 100.

A computer screen shot of a random text

Description automatically generated

**Correlation:**

It is a process of extraction a particular value from the response body and use it in the subsequent response using “Regular Expression Extractor” under post-processor.

**How to add it:**

Right click on Test plan 🡪 Add a HTTP Request sampler to hit the server for login 🡪 right click again 🡪 add post-processor 🡪 Regular Expression Extractor.

**How will it work:**

It has few fields associated to it and user must update as follows.

Name of the created variable 🡪 some random variable name

Regular Expression 🡪 <a href=”(\*)”>

Note: use regexp.com website, parse to the element like find an xpath where bracket indicates group and \* indicates any characters.

Template: $1$ // where 1 indicate in which group the value is available

Match No: 2 // at which index (starts from 1) the value is available when there are more than one

Default value: // left blank

8+ yrs exp

Cts, cmp1 and cmp 3

Agile methodology, 2 weeks,

Hp client -> api for validation, postman tool for api validation - > jmeter

2 weeks sprint

Can we give story point = 13?