

#503(prac4a)---27/07/19---

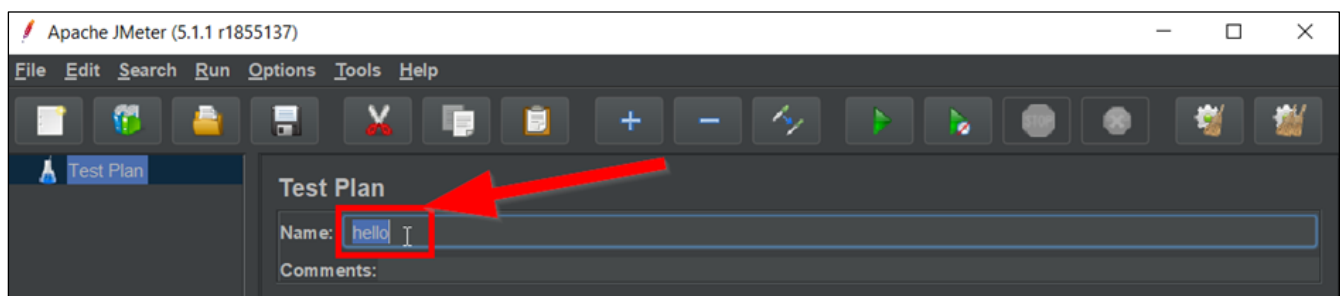
#AIM: Load Testing using JMeter.

PRE-REQUISITES:

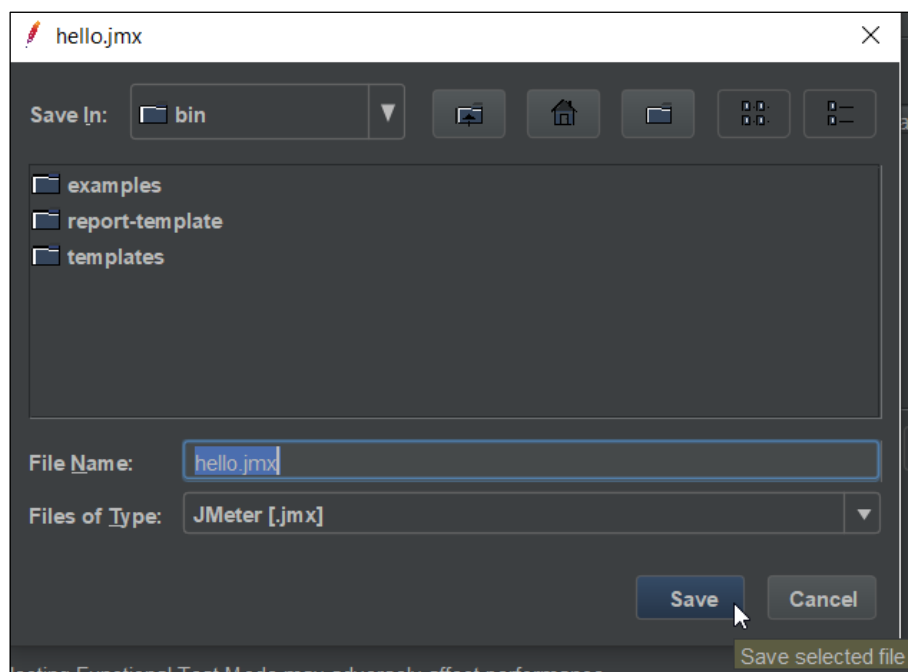
- 1) To Download “JDK”:
 - Visit <https://www.oracle.com/technetwork/java/javase/downloads/jdk12-downloads-5295953.html>
 - Download this file “jdk-12.0.2_windows-x64_bin.exe” and install it.
- 2) To Download “Apache JMeter”:
 - Visit https://jmeter.apache.org/download_jmeter.cgi
 - Under section “Apache JMeter 5.1.1 (Requires Java 8+)”
 - Under sub-section “Binaries”, download “apache-jmeter-5.1.1.zip” file.
 - Installation:
 - Extract the “apache-jmeter-5.1.1.zip” file.
 - Navigate to: **apache-jmeter-5.1.1 > bin > the “ApacheJMeter.jar” file** is your working space.

STEPS:

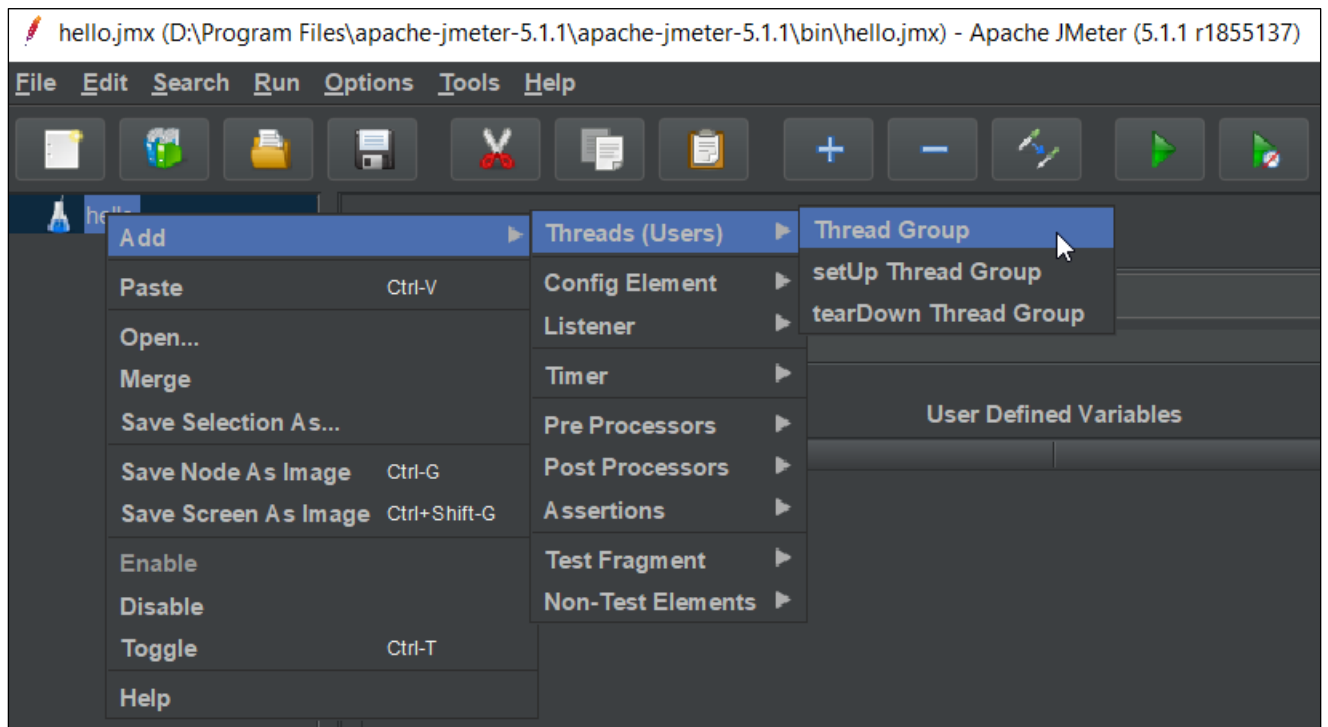
- 1) Open “ApacheJMeter.jar” file. Name your *Test Plan* as “hello”:



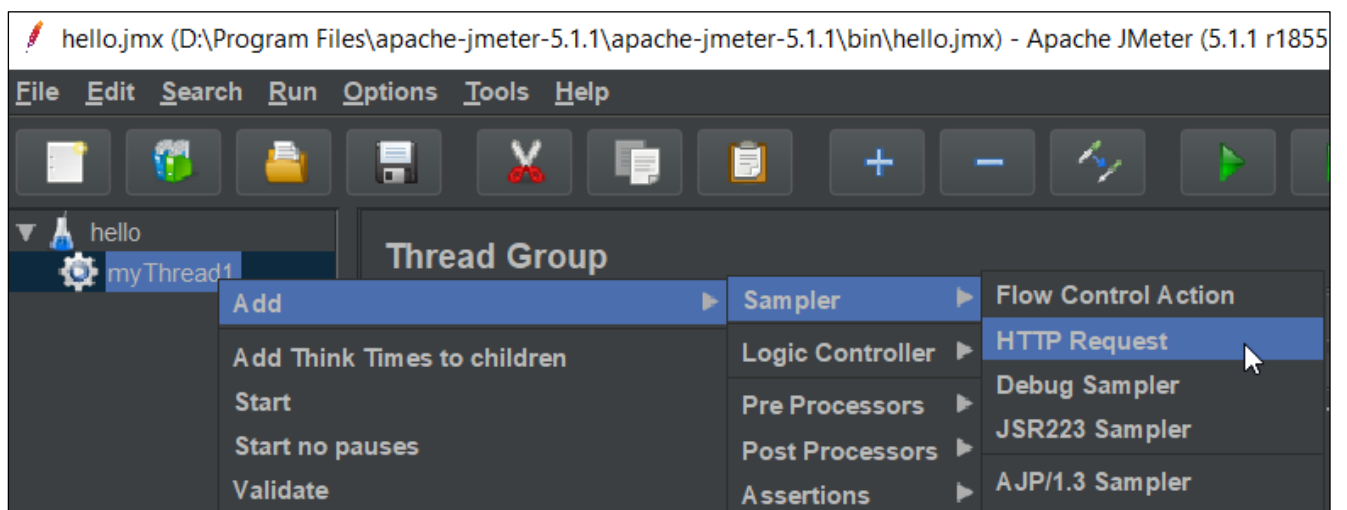
- 2) Save(CTRL+S) your Test Plan. It will be saved as “.jmx”(i.e. Java Management Extensions):



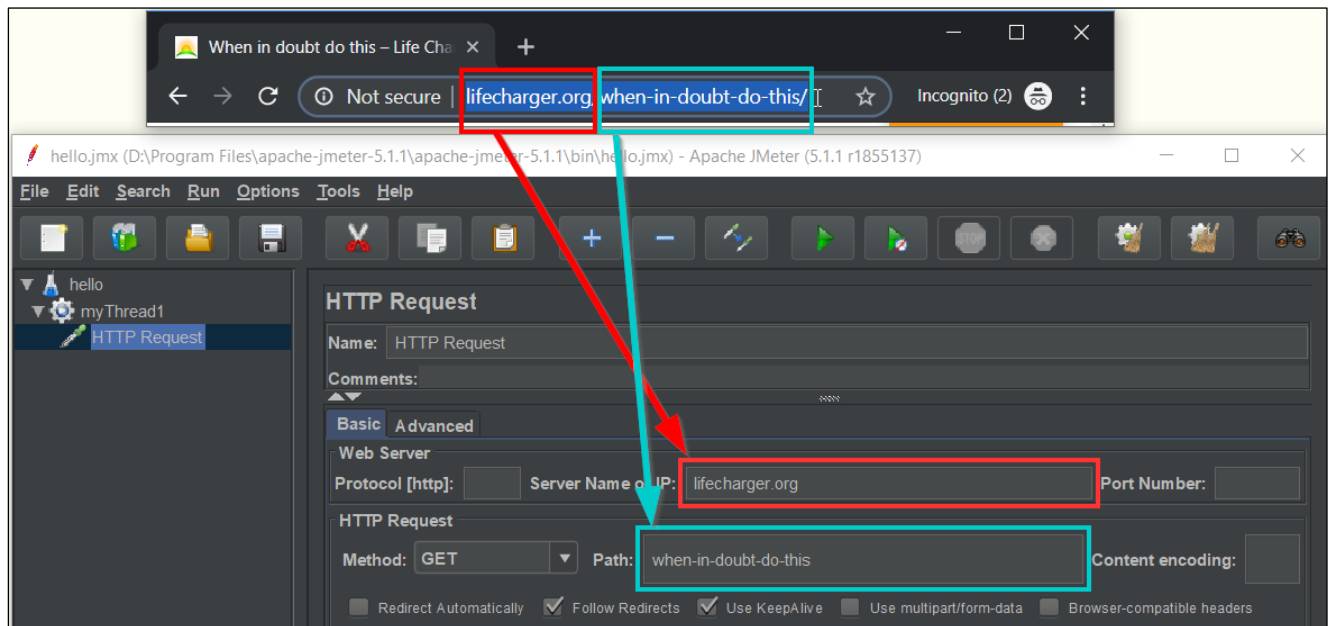
- 3) Create a **Thread Group**
(right-click over “hello(Test Plan)” > Add > Thread(Users) > Thread Group >
Name your **Thread Group** as “myThread1” > Save(CTRL+S)):



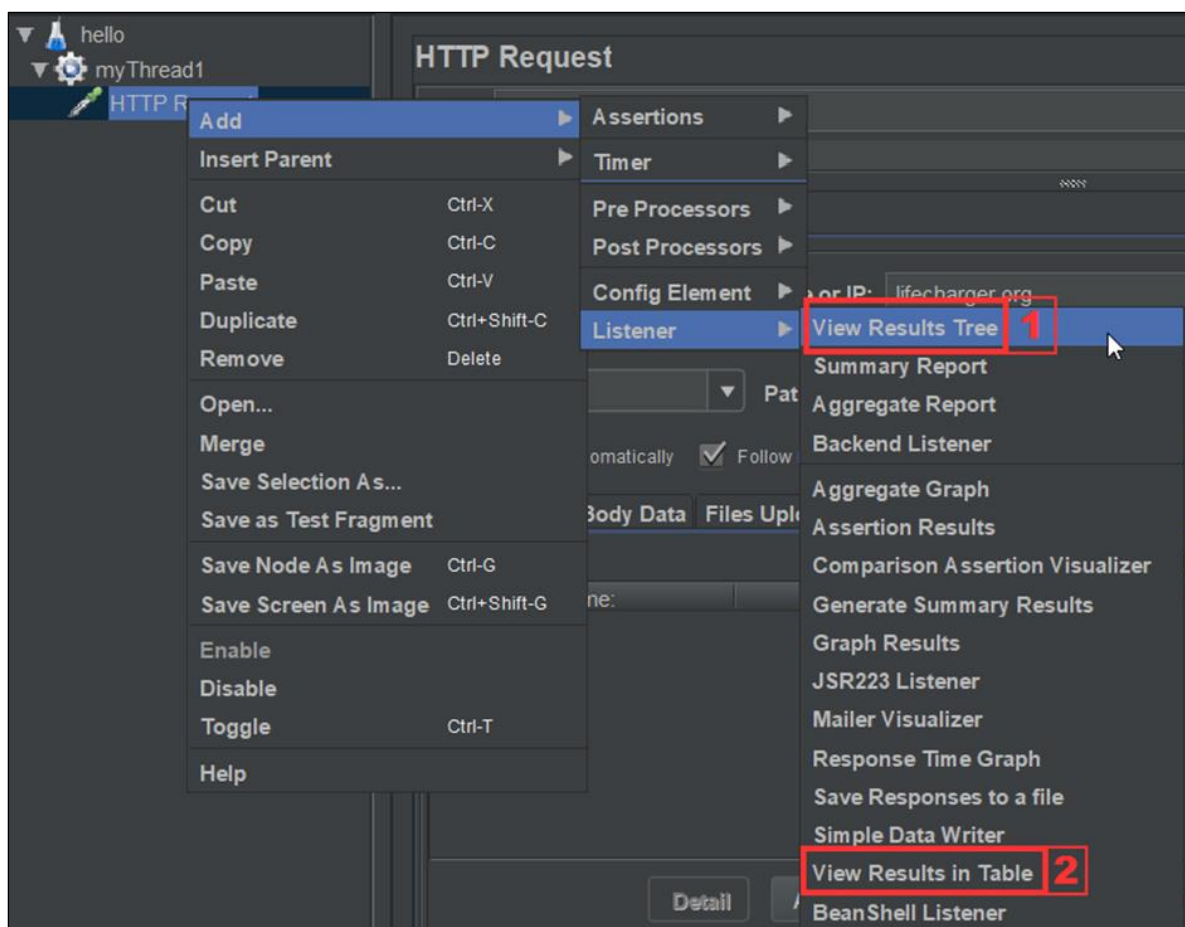
- 4) Under Thread Group(myThread1), add a “Sampler”, namely **HTTP Request**:
(right-click over “myThread1(Thread Group)” > Add > Sampler > HTTP Request):



- 5) Now visit any website page with a next path. Then **set website name** as “**Server Name or IP**” & **next path** as “**Path**” in the Basic panel:



- 6) Under HTTP Request Sampler, add 2 “Listeners”, namely **View Results Tree** & **View Results in Table**:
(right-click over HTTP Request(Sampler) > Add > Listener > View Results Tree) &
(right-click over HTTP Request(Sampler) > Add > Listener > View Results in Table):



7) **Save(CTRL+S) & Run(CTRL+R)** the file and wait for a while. You'll see some:

- **OUTPUTS:**

The screenshot shows the JMeter interface with the 'View Results Tree' panel selected. The left sidebar shows a project named 'hello' with a thread group 'myThread1' containing an 'HTTP Request' sampler. The 'View Results Tree' panel displays the 'Sampler result' for the 'HTTP Request'. A red box highlights the 'HTTP Request' in the tree, and a red arrow points to the 'Sampler result' tab. A cyan box highlights the 'Sampler result' details, which include:

- Thread Name: myThread1 1-1
- Sample Start: 2019-08-03 00:08:06 IST
- Load time: 13052
- Connect Time: 4897
- Latency: 8178
- Size in bytes: 42516
- Sent bytes: 275
- Headers size in bytes: 1907
- Body size in bytes: 40609
- Sample Count: 1
- Error Count: 0
- Data type ("text"/"bin"/""): text
- Response code: 200
- Response message: OK

Below these details, the 'HTTPSampleResult fields' are listed:

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

A cyan callout bubble labeled 'OUTPUT' points to the 'Sampler result' details.

The screenshot shows the JMeter interface with the 'View Results in Table' panel selected. The left sidebar shows the same project structure. The 'View Results in Table' panel displays a table with the following data:

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	00.08.06.323	myThread1 1-1	HTTP Request	13052	Success	42516	275	8178	4897

A cyan callout bubble labeled 'OUTPUT' points to the table.

8) Now change some Thread Properties(**users=10; ramp-up period=20; loop count=Forever**):

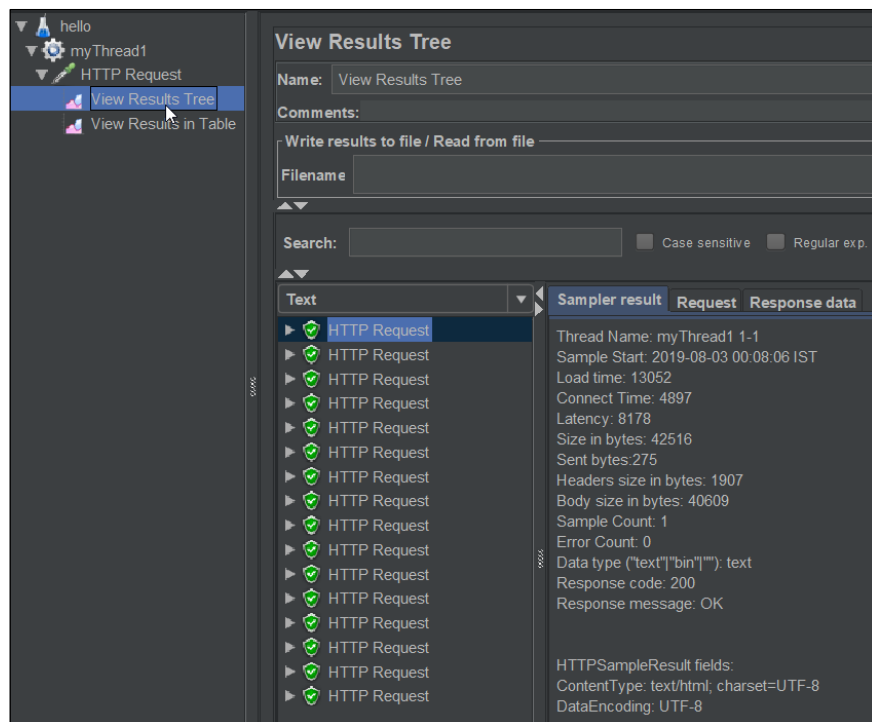
The screenshot shows the JMeter interface with the 'Thread Group' configuration panel selected. The left sidebar shows the project structure. The 'Thread Group' panel displays the following configuration:

- Name: myThread1
- Comments:
- Action to be taken after a Sampler error: Continue
- Thread Properties:
 - Number of Threads (users): 10
 - Ramp-Up Period (in seconds): 20
 - Loop Count: ☒ Forever

A red box highlights the 'Thread Properties' section.

9) Save(CTRL+S) & Run(CTRL+R) the file and wait for a while. You'll see some infinite:

- **OUTPUTS:**



Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
16	00:27:10.416	myThread1 1-1	HTTP Request	16990	Success	42516	275	6990	252
17	00:27:12.420	myThread1 1-5	HTTP Request	17123	Success	42495	275	7993	269
18	00:27:14.538	myThread1 1-6	HTTP Request	16339	Success	42516	275	7890	287
19	00:27:15.436	myThread1 1-2	HTTP Request	16516	Success	42516	275	7993	259
20	00:27:18.433	myThread1 1-3	HTTP Request	16993	Success	42495	275	7965	258
21	00:27:18.910	myThread1 1-7	HTTP Request	17281	Success	42516	275	8210	278
22	00:27:22.423	myThread1 1-8	HTTP Request	15061	Success	42516	275	6071	259
23	00:27:25.417	myThread1 1-9	HTTP Request	16071	Success	42524	275	6014	259
24	00:27:26.538	myThread1 1-10	HTTP Request	15378	Success	42517	275	5933	262
25	00:27:25.504	myThread1 1-4	HTTP Request	16518	Success	42516	275	5990	257
26	00:27:27.407	myThread1 1-1	HTTP Request	15503	Success	42516	275	7025	260
27	00:27:29.543	myThread1 1-5	HTTP Request	15941	Success	42517	275	7864	260
28	00:27:30.878	myThread1 1-6	HTTP Request	17631	Success	42523	275	8538	270
29	00:27:30.953	myThread1 1-2	HTTP Request	18806	Success	42516	275	8482	257
30	00:27:35.426	myThread1 1-3	HTTP Request	17313	Success	42523	275	6191	251
31	00:27:36.192	myThread1 1-7	HTTP Request	18255	Success	42523	275	7242	255
32	00:27:37.485	myThread1 1-8	HTTP Request	18219	Success	42516	275	6967	259
33	00:27:41.916	myThread1 1-10	HTTP Request	16316	Success	42516	275	9552	267
34	00:27:42.023	myThread1 1-4	HTTP Request	16587	Success	42495	275	10409	262
35	00:27:41.488	myThread1 1-9	HTTP Request	17198	Success	42517	275	8962	263
36	00:27:42.911	myThread1 1-1	HTTP Request	17076	Success	42516	275	9585	259
37	00:27:48.510	myThread1 1-6	HTTP Request	12850	Success	42516	275	6909	258
38	00:27:45.485	myThread1 1-5	HTTP Request	16055	Success	42516	275	6979	264
39	00:27:49.760	myThread1 1-2	HTTP Request	13732	Success	42516	275	6637	257

10) Finish!

Apache JMeter is:

- designed to load test functional behavior and measure performance.
- used to test performance both on static and dynamic resources, web dynamic applications.

Some features include: ability to test many apps/server/protocol types; Test Plan recording, building, debugging; complete report; portable; pure java; multi-threading; offline analysis of results; highly extensible core.