1) Write a program to demonstrate how a JMeter script is recorded.

Configuring HTTP Script Recorder to your JMeter

- Open Jmeter by clicking on the JMeter icon on the desktop in your practice lab.
- Right click on Test Plan.
- Click on Thread(Users) -> Thread Group.

- Right click on Workbench.
- Click on Add -> NonTest Elements -> Http(s) Test Script Recorder.

- Right click on Thread Group.
- Click on Add -> Logic Controller -> Recording Controller.

- Right click on Thread Group.
- Click on Add -> Listener -> View Result in Tree.

Step 1.1.2: Doing proxy settings in JMeter

- Click on HTTP(S) Script Recorder.
- Set port number (optional).
- Click on the Start Button.

Note: The Default Certificate will be generated in your local machine.

Step 1.1.3: Configuring JMeter in browsers (using Firefox browser here)

- Start Firefox, but do not close JMeter.
- From the toolbar, click Edit → Preferences (or Tools → Preferences or type about:preferences#advanced as the URL). This should bring up the options.

- Select the Advanced tab and Network tab.
- Click Settings Button near the top.

- On the new POP-UP, check Manual Proxy Configuration. The address and port fields should be enabled now.

  Address:

  Enter localhost or the IP address of your system.

  Port:

  Enter 8181(Samp port in HTTP(S) Script Recorder)

- Check the Use this proxy server for all protocols checkbox.
- Click the Ok button and that should return you to the browser.

Step 1.1.4: Recording the script

- Open the website you want to check in a browser, example: Firefox.
- Recording Controller records the script.

- View the results in View Result Tree.

2) Write a program to demonstrate how load (threads) is added in JMeter.

Adding thread load

- **Open JMeter.**
- **Right click on Test Plan.**
- **Click on Thread (Users) -> Concurrency Thread Group.**

- **Right click on Workbench.**
- **Click on Add -> Non Test Elements -> HTTP(S) Test Script Recorder.**

- **Right click on Thread Group.**
- **Click on Add -> Logic Controller -> Recording Controller.**

- Right click on Concurrency Thread Group.
- Click on Add -> Listener -> View Result Tree.

Step 1.2.2: Doing proxy settings in JMeter

- Click on HTTP(S) Script Recorder.
- Set port number (optional).
- Click on the Start button.

Note: A Default Certificate will be generated in your local machine.

Step 1.2.3: Configuring JMeter in browsers (using Firefox browser here)

- Start Firefox, but do not close JMeter.
- From the toolbar, click Edit → Preferences (or Tools → Preferences or type about: preferences#advanced as the URL). This should show the options.

- Select the Advanced tab and Network tab.
- Click on Settings.
- On the new pop-up screen, check Manual Proxy Configuration. The address and port fields should be enabled now.

  Address:

    Enter localhost or the IP address of your system.

  Port:

    Enter 8181 (Samp port in HTTP(S) Script Recorder).

- **Check the Use this proxy server for all protocols checkbox.**
- **Click on the Ok button and this should return you to the browser.**

Step 1.2.4: Demonstrating an example of multi thread load

- **Right click on the Concurrency thread group.**
- **Add targeted threads (number threads).**
- **View the result of adding multiple thread loads.**

- **Run any website in the Firefox browser and see the end result.**
- **View the end results in View Result Tree.**

3) Write a program to demonstrate how load is increased in JMeter.

Increasing the load

- **Open JMeter.**
- **Right click on TestPlan.**
- **Click on Thread(Users)->Concurrency Thread Group.**

Step 1.3.2: Demonstrating an example of increasing the load

- **Minimum ramp up of 5 (increasing the load by step count)**

- **Maximum ramp up of 10 (increasing the load by step count)**

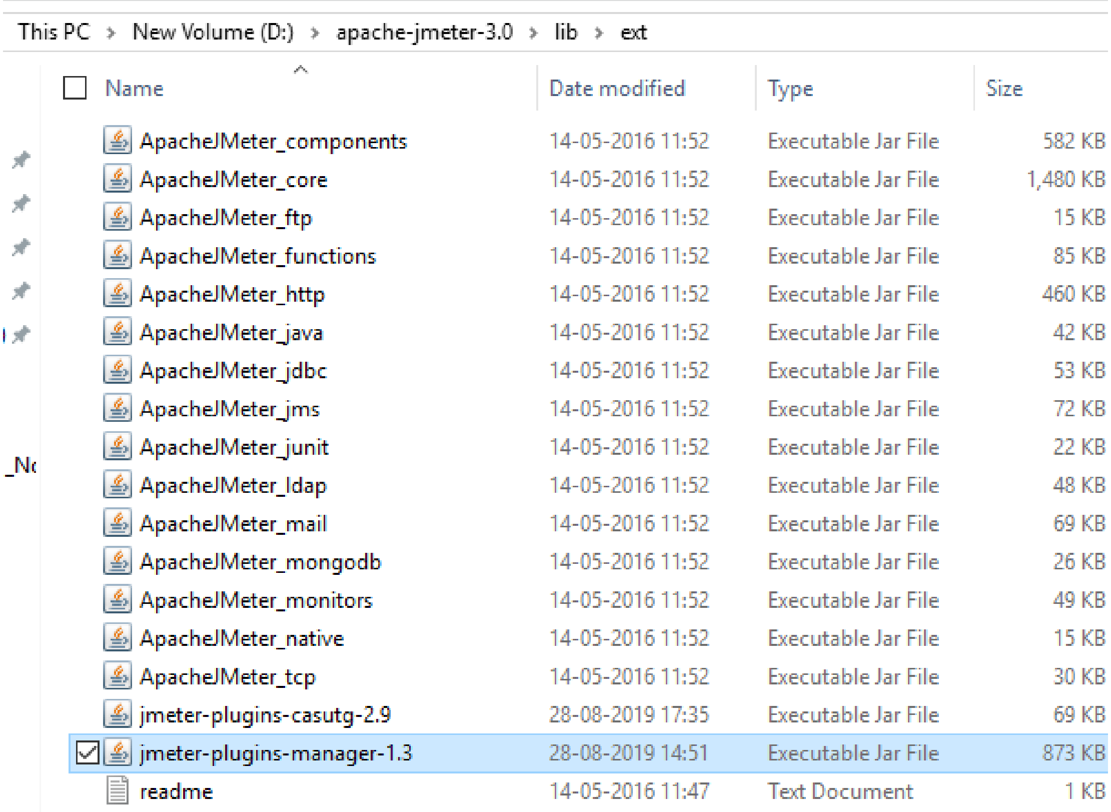
4) Write a program to demonstrate the use of thread group plugin in JMeter.

: Adding thread group plugin

- Go to any browser and type JMeter plugins manager.

- Click on the link https://jmeter-plugins.org.
- Now, once you click on **Install**, you will get a jar file that will install the plugins.
- The easiest way to get the plugins is to install the **Plugins Manager**.
- Now, we'll be able to install any other plugin just by clicking a checkbox. Link:- [Download plugins-manager.jar and put it into lib/ext directory, then restart JMeter.]()

- The downloaded plugins_manager.jar file will look like:



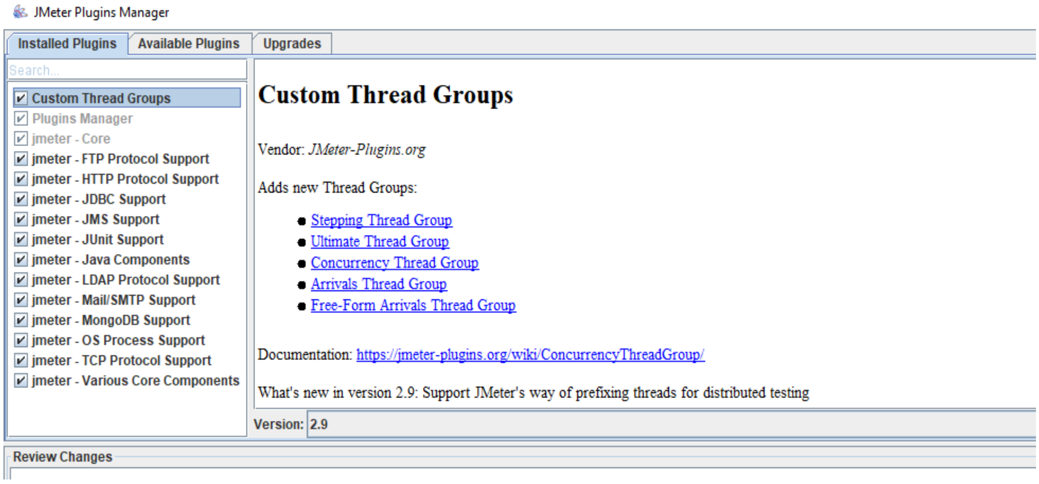
This PC › New Volume (D:) › apache-jmeter-3.0 › lib › ext

□ Name	Date modified	Type	Size
ApacheJMeter_components	14-05-2016 11:52	Executable Jar File	582 KB
ApacheJMeter_core	14-05-2016 11:52	Executable Jar File	1,480 KB
ApacheJMeter_ftp	14-05-2016 11:52	Executable Jar File	15 KB
ApacheJMeter_functions	14-05-2016 11:52	Executable Jar File	85 KB
ApacheJMeter_http	14-05-2016 11:52	Executable Jar File	460 KB
ApacheJMeter_java	14-05-2016 11:52	Executable Jar File	42 KB
ApacheJMeter_jdbc	14-05-2016 11:52	Executable Jar File	53 KB
ApacheJMeter_jms	14-05-2016 11:52	Executable Jar File	72 KB
ApacheJMeter_junit	14-05-2016 11:52	Executable Jar File	22 KB
ApacheJMeter_ldap	14-05-2016 11:52	Executable Jar File	48 KB
ApacheJMeter_mail	14-05-2016 11:52	Executable Jar File	69 KB
ApacheJMeter_mongodb	14-05-2016 11:52	Executable Jar File	26 KB
ApacheJMeter_monitors	14-05-2016 11:52	Executable Jar File	49 KB
ApacheJMeter_native	14-05-2016 11:52	Executable Jar File	15 KB
ApacheJMeter_tcp	14-05-2016 11:52	Executable Jar File	30 KB
jmeter-plugins-casutg-2.9	28-08-2019 17:35	Executable Jar File	69 KB
☑ jmeter-plugins-manager-1.3	28-08-2019 14:51	Executable Jar File	873 KB
readme	14-05-2016 11:47	Text Document	1 KB

Step 1.4.2: Adding a specific thread group plugin

- Open the JMeter tool.

- Click on the **Options** tab and then click on Plugins Manager.
- Search for the custom thread group plugin and check the box next to it.
- Finally, click on **Install** and restart the JMeter tool.



☑ Custom Thread Groups
☑ Plugins Manager
☑ jmeter - Core
☑ jmeter - FTP Protocol Support
☑ jmeter - HTTP Protocol Support
☑ jmeter - JDBC Support
☑ jmeter - JMS Support
☑ jmeter - JUnit Support
☑ jmeter - Java Components
☑ jmeter - LDAP Protocol Support
☑ jmeter - Mail/SMTP Support
☑ jmeter - MongoDB Support
☑ jmeter - OS Process Support
☑ jmeter - TCP Protocol Support
☑ jmeter - Various Core Components

Custom Thread Groups

Vendor: *JMeter-Plugins.org*

Adds new Thread Groups:

- Stepping Thread Group
- Ultimate Thread Group
- Concurrency Thread Group
- Arrivals Thread Group
- Free-Form Arrivals Thread Group

Documentation: https://jmeter-plugins.org/wiki/ConcurrencyThreadGroup/

What's new in version 2.9: Support JMeter's way of prefixing threads for distributed testing

Version: 2.9

5) Write a program to demonstrate how a SMTP sampler script is run in JMeter.
Adding javax.mail.jar file in JMeter

- **Javax.mail.jar file is already present in the /usr/share/java directory.**
- **Add javax.mail.jar file in JMeter libs directory.**
- **Restart JMeter.**

Step 1.5.2: Configuring the SMTP sampler script to send emails

- **Open JMeter.**
- **Right click on Test Plan.**
- **Go to Add ->Threads ->Thread Group.**

- **Right click on Thread Group.**
- **Now, click on Add ->sampler ->SMTP sampler.**

Step 1.5.3: Adding data in the SMTP sampler

- Server: smtp.googleemail.com (or) smtp.gmail.com
- Port: The default ports given are: 587, 25, or 465.
- Email Address: We can use any Gmail address for this (for example: Yarramadhu.209@gmail.com).
- The google mail SMTP server requires authentication, so select the “Use Auth” box.
- Username: Same as the email address (for example: Yarramadhu.209@gmail.com)
- Password: Enter the password (The password is displayed with asterisks in the JMeter GUI but it’s stored unencrypted (as is) in the JMeter.jmx file).
- Google mail SMTP server requires StartTLS, so click on StartTLS.
- Message settings: Leave this area as it is or fill it out in accordance with your test scenario.
- Subject: You can put anything you want here, leave it blank or send it without a subject by selecting it. But for now, we will write “Hello JMeter SMTP sampler” in the subject.
- Message: We will need to include the message body. This section accepts plain text and JMeter variables, functions, properties, and more.
- The ThreadNum function represents the number of thread currently being executed.

Step 1.5.4: Sending emails

- Click on Run and an email will be sent to your email address.

- Now, you will get an email but it will show a security alert.
- You can overcome this alert by going to Gmail settings.

- **Since the less secure app access is in off mode, we wouldn’t get mails.**
- **Therefore, you will have to turn it on.**

- **Once its turned on, we will start getting emails.**

- **The script will now run successfully.**
- **All the five emails will show up in your Gmail Web UI and the thread numbers will be displayed in the body preview.**

6) Write a program to demonstrate how a JDBC sampler script is run in JMeter.

Setting up a database (with database name, username, password, and port)

- **Open your database.**
- **Create a database and then create a table.**

Step 1.6.2: Downloading database specific jar and placing it in lib

- **mysql-connector-java jar file is already present in your practice lab.**

- **Add that jar file in the JMeter bin folder.**
- **Now, restart your JMeter.**

Step 1.6.3: Setting up the JDBC Config Manager

- **Open JMeter.**
- **Right click on Test Plan.**
- **Click on Add -> Thread (Users) -> Thread Group.**

- Right click on the Thread Group.
- Click on Add -> Config Element -> JDBC Connection Configuration.

- Enter your database URL.
- Enter your JDBC Driver Class (based on your custom database).
- Enter your username and password.

Step 1.6.4: Adding and configuring the JDBC Sampler

- Right click on Thread Group.
- Click on Add -> Sampler -> Configure JDBC Sampler.

- Enter your variable name which should be the same as the JDBC Connection Configuration.

Step 1.6.5: Writing the queries and viewing the result

- Right click on Thread Group.
- Click on Add -> Listener -> View Results Tree.

- Write the queries in JDBC Request.

- View the result in View Results Tree.
- The sample request result will look like:

- Click on Request to show queries.

- Click on Response data to show queries output.

7) Write a program to demonstrate how a JUnit sampler script is run in JMeter.
Creating a JUnit test project

- **We will create a sample java project having JUnit annotations.**
- **It contains a test class sampleJUnitTest.java having two test methods.**

Note: For the demo, we have two JUnit tests in sampleJUnitTest.java which are sampleTestPassing and sampleTestFailing. The test sampleTestPassing gets passed when you run it and the test sampleTestFailing explicitly fails when it uses the Assert.fail() command.

Step 1.7.2: Creating jar for the JUnit project

- **Right click on the project and click on the Export button.**

- **Inside java, click on the 'JAR' file.**

- **Select your project and check the resources. Also, provide the path for the generated 'jar' file.**

Step 1.7.3: Putting jar in JMeter's lib/JUnit directory

- **Put the generated jar file in the JMeter's lib/JUnit directory and restart JMeter.**

Step 1.7.4: Running JUnit tests in JMeter

- First, add a “JUnit Request” to a thread group.

- Check the “search for JUnit 4 annotations (instead of JUnit3)” checkbox.
- From the “classname” dropdown, select the JUnit test class created.
- From the “Test Method” dropdown, select the JUnit method/test where you want to load the test.
- Likewise, multiple JUnit requests can be added with each request having a test method.
- In this example, two JUnit requests are added for passing and failing the test.

- Add Listeners and run the test.

8) You are given a project to demonstrate load and scalability with JMeter.

Problem statement for Load & Scalability Testing in JMeter

- Objective: Create Load & Scalability in JMeter for a particular domain.
- Steps involved:

1) Set Up JMeter.

2) Create a Test Plan.

3) Add Thread group.

4) Record the Scripts.

5) Add Https Cookies Manager.

6) Add Listener and view the results.

Step 1.1.2 Solution for the problem statement

1) Set up a Thread Group:

- **Right-click on Thread Group.**
- **Click on Add ->Thread Users -> Thread Group.**

- **Add the number of Thread Users.**
- **Add a ramp-up period.**
- **Add a loop count.**

2) Add an HTTP Request:

- **Right-click on Thread Group.**
- **Click on Add -> Sampler -> HTTP Request.**

- **Enter the IP Address.**
- **Enter the path.**

3) Add the HTTP Cookies Manager:

- Right-click on Thread Group.
- Click on Add -> Config Element -> HTTP Cookie Manager.

- Click on Add.
- Check if there is a credentials page.
- Enter the username and password.

4) Add Listener to view the results:

- Right-click on Thread Group.
- Click on Add -> Listener -> View the result.

- Run the Scripts and view the results.

9) Write a program to demonstrate how Selenium WebDriver is integrated using JMeter script.

Development Environment:

- Apache JMeter 5.1.1 Version
- OpenJDK Runtime Environment 11.0.2

This guide has five sub-sections, namely:

1.8.1 Installing Plugins Manager and configuring it to JMeter

1.8.2 Installing WebDriver plugin

1.8.3 Writing a WebDriver Script

1.8.4 Running WebDriver tests in JMeter

1.8.5 Pushing the code to your GitHub repositories

Steps 1.8.1: Installing Plugins Manager and configuring it to JMeter

- Download WebDriver plugins for JMeter.
- Download the JMeter Plugin from: “https://jmeter-plugins.org/install/Install/”.

- Unzip JMeterPlugins-WebDriver 1.1.3 (version is default).
- The unzipped files must be copied into the lib folder -> ext folder of JMeter.

Steps 1.8.2: Installing WebDriver plugin

- Select Options --> click on Plugins Manager.

- Click on the Available Plugins. Now, click on the Selenium/WebDriver Sampler.
- Click on Restart and the Install button. (JMeter will automatically restart).
- Click on install plugins and then click on Selenium/WebDriver Sampler.

Steps 1.8.3: Writing a WebDriver Script

- Right click on Test Plan.
- Click on Add -> Thread(Users) -> Thread Group.

- Right click on Thread Group.
- Click on Add -> Config Element -> jp@gc Chrome Driver Config.
- You can also use the Firefox Driver for this.
- Click on Add ->Config Element ->jp@gc Firefox Driver Config.

- **Right click on Thread Group.**
- **Click on Add -> Sampler -> jp@gc WebDriver Sampler.**

- **Right click on Thread Group.**
- **Click on Add -> Sampler -> Listener-> View Results Tree.**

Steps 1.8.4: Running WebDriver tests in JMeter

- **Click on jp@gc Chrome Driver Config.**
- **Enter your Driver path.**
- **You can also use Firefox Driver Config.**
- **Enter your Driver path.**

- **Click on jp@gc WebDriver Sampler.**
- **Write the Selenium script.**

- **Run the script.**

- View the results in View Result Tree.

10) Write a program to demonstrate a WebDriver set sampler script in JMeter.

: Installing JMeter Plugins Manager

- Go to Google and search for 'JMeter Plugins Manager.'
- Download the Plugin Manager jar file.
- Unzip the jar file and add that jar file to the lib folder.

- Restart JMeter.

Step 1.9.2: Adding the WebDriver Set Sampler

- Right click on Test Plan.
- Click on Add -> Thread Users -> Thread Group.

- Right click on Thread Group.
- Click on Add -> Config Element -> jp@gc - Firefox driver Config.

- Add Config Element -> HTTP Cookie Manager.
- Select Config Element -> jp@gc - Firefox Driver Config.
- Select Sampler ->jp@gc -Web Driver Sampler.
- Select Listener -> View Results Tree

Step 1.9.3: Running the scripts in WebDriver Set Sampler

- **Open the WebDriver Sampler.**
- **Run the script.**

11) Demonstrate a FTP sampler in JMeter.

: Adding users in the thread group

- **Open JMeter.**
- **Right click on Test Plan.**
- **Select Add → Thread Group.**

Steps 1.10.2: Adding Default FTP Request Properties

- **Right click on Test Plan.**
- **Select Add → Config Element → FTP Request Defaults.**

Steps 1.10.3: Adding FTP Request

- **Select Add → Sampler → FTP Request.**

- **Change the name to 'File1.'**
- **Change the Remote File field to '/directory/file1.txt.'**

- **Change the Username field to ‘anonymous.’**
- **Change the Password field to ‘anonymous@test.com.’**

Steps 1.10.4: Adding Listener and viewing the results in a table

- **Right click on Test Plan.**
- **Select Add → Listener → View Results in the table.**

12) Write a program to demonstrate OS process, debug and test action sampler script in JMeter.

Creating an OS process in JMeter

- **Open JMeter.**
- **Right click on Test Plan.**
- **Click on Thread(Users)->Thread Group.**

- **Right click on the thread group.**
- **Go to Add -> Sampler -> OS process sampler.**

- **Right click on the thread group.**
- **Go to Add -> Listener -> View results tree.**

- **Run the result tree.**

Steps 1.11.2: Debugging in JMeter and testing the action sampler

- **Right click on Test Plan.**
- **Click on Thread(Users) -> Thread Group.**
- **Right click on thread group -> Add -> samplers -> HTTP Request -> Enter Server name 'lifecharger.org.'**
- **Run the result tree.**
- **Click on the drop down button and debug particular elements like HTML/CSS file depending on your requirement.**
- **The text action sampler will have multiple thread groups in the Test Plan. The second Thread Group should start execution only a few seconds after the 1st Thread Group gets started.**
- **Right click on Test Plan.**
- **Click on Thread(Users) -> Add two Thread Groups.**
- **Right click on the thread group.**
- **Go to Add -> Samplers -> Add two HTTP Request -> Thread 1.**
- **Now, enter the server name as 'lifecharger.org.'**
- **Thread 2 -> Enter Server name 'www.facebook.com.'**

Note: This can be achieved by ramp_up delay between the two threads.

- **Right click on the thread group.**

- Go to Add -> Listener -> View Results Tree.

- Run the result tree.

13) Write a program to demonstrate random and random order controllers in JMeter.

Recording scripts

- Open JMeter.

- Right click on Test Plan.

- Click on Thread(Users) -> Thread Group.

- Right click on WorkBench.

- Click on Add -> Non Test Elements ->HTTP(s) Test Script Recorder.

- Right click on Thread Group.

- Click on Add -> Logic Controller->Recording Controller.

- **Right click on Thread Group.**
- **Click on Add -> Listener->View Results Tree.**

- **Configure proxy settings in the browser.**
- **Run the scripts.**
- **Automatically JMeter records the scripts.**

Step 1.12.2: Adding random controller

- **Right click on the Recording Controller.**
- **Click on Add -> Logic Controller -> Random Controller.**

Steps 1.12.3: Running the scripts in random controller

- **Recording scripts should be added as child scripts for the random controller.**
- **In a random controller, it will select only one script randomly and execute the selected script.**

- After execution, in the output screen, it shows only one script which has been randomly picked by the random controller.

Steps 1.12.4: Adding random order controller

- Right click on the Recording Controller.
- Click on Add -> Logic Controller -> Random Order Controller.

Step 1.12.5: Running the scripts in random order controller

- Recording scripts should be added as child scripts for the random order controller.
- In the random order controller, all the child scripts will be selected. By doing this, it executes all the scripts randomly and the output will also be in a random order.

14)Write a program to demonstrate interleave controller in JMeter.

Recording the Scripts

- Open JMeter.
- Right click on Test Plan.
- Click on Thread (Users) -> Thread Group.

- Right click on WorkBench.
- Click on Add -> Non Test Elements ->HTTP(s) Test Script Recorder.

- Right click on Thread Group.
- Click on Add -> Logic Controller->Recording Controller.

- Right click on Thread Group.
- Click on Add -> Listener->View Result in Tree.

- Configure proxy settings in the browser.
- Run the scripts.
- JMeter automatically records the scripts.

Step 1.13.2: Adding interleave controller

- Right click on Thread Group.
- Click on Add -> Logic Controller -> Interleave Controller.

Steps 1.13.3: Run the scripts in interleave controller

- Recorded scripts should be added as child scripts for the interleave controller.

- Thread group hasan option called loop count in which you can give the loop count value.

- Based on the loop count value, the output is available at that time of execution. The output will be available based on the inserted order of the interleaving controller.

- If you don’t change the loop value and try to run it frequently, you will get the equivalent output as per the result of the first execution.

15) Write a program to demonstrate throughput controller in JMeter.

Loading distribution of users

- Let’s take an example of “www.lifecharger.org” website.

- This website contains some pages like: Home page,About page and Archives page.

Step 1.14.2: Adding throughout controller

- Right click on Thread Group.

- **Click on Add -> Logic Controller ->Throughput Controller.**

- **Right click on the Throughput Controller.**
- **Click on Add -> Sampler -> HTTP Request.**

- **Rename HTTP Request as HomePage.**
- **Give an IP address.**
- **Path should be kept empty because the website landing page is homepage.**

- **Create another throughput controller.**
- **Add HTTP request and rename as About Page.**

- **Give an IP address.**

- Fill the Path box.

- Create another throughput controller.
- Add HTTP request and rename it as Archive Page.

- Give an IP Address.
- Fill the Path box.

- Right click on Thread Group.
- Click on Add -> Listener -> Aggregate graph.

Step 1.14.3: Performing total execution testing in throughout controller

- The number given for throughput will be the total number of executions for the current controller.

- In Thread Group, consider 10 concrete users and the loop count as 1 (concrete users optional).

- Throughput Controller should be in Total Executions.
- Click on the check box “per user”.
- Create three throughput controllers as shown below.
- Create Throughput Controller 1 and give ‘Throughput’ as 2.

- Next, create Throughput Controller 2 and give ‘Throughput’ as 3.

- Again, create Throughput Controller 3 and give ‘Throughput’ as 5.

- Run the throughput controller and view the results as shown below:

Step 1.14.4: Performing percentage execution testing in throughout controller

- In Thread Group, consider 10 concrete users and loop count 1 (concrete users optional).

- **Throughput Controller should be in Percentage Executions.**
- **Throughput Controller is optional.**
- **Uncheck the checkbox “per user”.**
- **Create three Throughput Controllers as shown below.**
- **Create Throughput Controller 1 and consider 20% as Throughput.**

- **Create Throughput Controller 2 and consider 30% as Throughput.**

- **Create Throughput Controller 3 and consider 50% as Throughput.**

- **Run the Throughput Controller and view the results as shown below:**

16) Write a program to demonstrate runtime controller in JMeter.

: Recording scripts

- **Open JMeter.**
- **Right click on Test Plan.**
- **Click on Thread(Users) -> Thread Group.**

- **Right click on Work Bench.**
- **Click on Add -> Non Test Elements ->HTTP(s) Test Script Recorder.**

- **Right click on Thread Group.**
- **Click on Add -> Logic Controller->Recording Controller.**

- **Right click on Thread Group.**
- **Click on Add -> Listener->View Result in Tree.**

- **Configure proxy settings in the browser.**
- **Run the scripts.**
- **Automatically JMeter records the scripts.**

Step 1.15.2: Adding runtime controller

- **Right click on Thread Group.**
- **Click on Add -> Logic Controller -> Runtime Controller.**

Step 1.15.3: Running the scripts in runtime controller

- **Recording scripts should be added as child scripts for the runtime controller.**
- **Provide the runtime and run the scripts.**

- **Run the throughput controller and view the results as below:**

17) Write a program to demonstrate if and while controllers in JMeter. : Recording scripts

- **Open JMeter.**
- **Right click on Test Plan.**
- **Click on Thread Group -> Thread Users -> Thread Group.**

- **Rename Thread Group as “Thread Group before If Controller”.**
- **Right click on “Thread Group before If Controller.**
- **Click on Add -> Sampler ->HTTP Request.**

- **Rename HTTP Request as “HTTP Request before If controller”.**
- **Provide an IP address and path.**

Step 1.16.2: Adding If controller

- **Right click on “Thread Group before If Controller”.**

- **Click on Add -> Logic Controller -> If Controller.**

- **Right click on “Thread Group Before If Controller”.**
- **Click on Add -> Sampler -> HTTP Request.**

- **Rename HTTP Request as “HTTP Request after If Controller”.**
- **Give the same IP address and path as HTTP Request before the if controller.**

Step 1.16.3: Running the scripts in If controller and viewing the results

- **Write the conditions in the If Controller.**

- **If the condition is True, then it runs the script.**

- If the condition is False, then it doesn't run the script.

Step 1.16.4: Adding While controller

- Right click on Test Plan.
- Click on Add -> Thread Users -> Thread Group.

- Right click on Thread Group.
- Click on Add -> Logic Controller -> While Controller.

- Right click While loop Control.
- Click on Add -> Sampler -> HTTP Request.

Steps 1.16.5: Adding CSV data set Config to While controller

- **Right click on While controller.**
- **Click on Add -> Config Element -> CSV Data Set Config.**

- **Create CSV file.**
- **Write scripts in CSV file.**

- **Browse the CSV file in CSV data set Config.**
- **Allow the quoted data, recycle and stop thread on EOF. (true or false optional).**

Steps 1.16.6: Running the scripts in While controller

- **Open HTTP Request and give the IP address as head of column in CSV file.**

- **Open While controller and input the function.**

- **Run the scripts and view the result.**

18) Write a program to demonstrate switch, module, and include controllers in JMeter.

Demonstrating Switch controller

- **Open JMeter.**
- **Right click on Test Plan-->Add-->Threads-->Thread Group.**

- **Right click on Thread-->Add-->Logic Controllers -->Switch Controller.**

- **Right click on Switch Controller-->Add -->Sampler-->HTTP Request Default.**

- **Right click on Switch Controller-->Add -->Sampler-->HTTP Request_1.**

- **Right click on Switch Controller-->Add -->Sampler-->HTTP Request _2.**

- **Right click on Test Plan--> Add-->Listener-->View Result Tree.**

Steps 1.17.2: Demonstrating the module controller

- **Right click on Test Plan-->Add-->Threads-->Thread Group.**

- **Right click on Thread Group-->Add -->Logic Controller-->Module Controller.**

- **Right click on Thread Group → Add → HTTP Request as shown below:**

- **Right click on Thread Group-->Add -->Logic Controller-->Simple Controller--> change name to start code.**

- **Right click on start code -->Add-->Sampler-->Add two HTTP Requests.**

- **Note: Now right click on start code and disable it. Click on the Calling start code module controller. Select start code file and run the thread group.**

- **Right click on Test Plan-->Add --> Listener-->View Result Tree-->Run.**

Step 1.9.3: Demonstrating include controller

- **Right click on Test Plan-->Add-->Threads-->Thread Group.**

- **Right click on Thread Group-->Add -->Sampler-->HTTP Request.**

- **Right click on Test Plan-->Add -->Test Fragment-->**

-->Add-->HTTP Request under Test Fragment.

- **Right click on Thread Group-->Add-->Logic Controller-->Include Controller-->load .jmx file in Include Controller.**

- **Right click on Test Plan-->Add --> Listener-->View Result Tree-->Run.**

19)Write a program to demonstrate ForEach controller with regular expression extractor in JMeter.

Step 1.18.5: Pushing the code to GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

```
cd <folder path>
```

Initialize your repository using the following command:

```
git init
```

Add all the files to your git repository using the following command:

```
git add .
```

Commit the changes using the following command:

```
git commit . -m “Changes have been committed.”
```

Push the files to the folder you initially created using the following command:

```
git push -u origin master
```

20) Write a program to demonstrate ForEach controller with Xpath extractor in JMeter.

Adding thread group

- **Open JMeter.**
- **Right click on Test Plan-->Add-->Threads-->Thread Group.**

- **Right click on Thread-->Add-->Sampler -->Xpath Extractor--> HTTP Request.**

Step 1.19.2: Adding For Each controller

- **Right click on Thread Group-->Add -->Logic Controllers-->For Each Controller.**

Step 1.19.3: Adding Xpath extractor and debug sampler.

- **Right click on Thread Group-->Add -->PostProcessor-->Add two Xpath Extractor with different values as shown in below:**

- **Right click on Thread Group-->Add -->Sampler-->Debug Sampler.**

- Right click on Test Plan--> Add-->Listener-->View Result Tree.

21) Write a program to demonstrate implementation of JMeter controllers using Selenium.

Problem statement for Processors, Assertions and Timers

- Objective: Implement Selenium with JMeter for a particular domain
- Steps involved:

1. Set up JMeter.
2. Create a test plan.
3. Add thread group.
4. Record the scripts.
5. Install WebDriver plugin.
6. Write Selenium scripts in WebDriver.
7. Run the scripts and view the results.

Step 1.2.2: Solution for the problem statement

1. Migrating plugins manager to JMeter.

- Download WebDriver plugins for JMeter.
- Download JMeter Plugin from “https://jmeter-plugins.org/install/Install/”.

- Unzip JMeterPlugins-WebDriver 1.1.3 (version is default).
- The unzipped files must be copied to the lib folder -> ext folder of JMeter.

2. Installing WebDriver plugin

- **Click on options -> click on plugins manager.**

- **Click on Available plugins -> click on Selenium/Web Driver Sampler.**
- **Click on the Restart and install button. (JMeter will automatically restart.)**
- **Click on install plugins -> click on Selenium/Web Driver Sampler.**
- **Click on the Restart and install button.(JMeter will automatically restart.)**

3. Writing your Selenium script.

- **Right click on the Test Plan.**
- **Click on Add -> Thread(Users)->Thread Group.**

- **Right click on Thread Group.**
- **Click on Add -> Config Element->jp@gc Chrome Driver Config.**

You can also use Firefox driver.

- **Click on Add ->Config Element ->jp@gc Firefox Driver Config.**

- **Right click on Thread Group.**
- **Click on Add -> Sampler->jp@gc Web Driver Sampler.**

- **Right click on Thread Group.**
- **Click on Add -> Sampler->Listener-> View Results in Tree.**

4. Running WebDriver tests in JMeter

- **Click on jp@gc Chrome Driver Config.**
- **Enter your Chrome driver path.**
- **You can also use Firefox Driver Config.**
- **Enter your Firefox driver path.**
- **Click on jp@gc Web Driver Sampler.**
- **Write your Selenium script**

- **Run the script.**
- **View the results in “View Results Tree”.**

22) Write a program to demonstrate pre and post-processors in JMeter.

Adding preprocessor

- **Right click on Test Plan-->Add-->Threads-->Thread Group.**

- **Right click on Thread-->Add-->PreProcessor -->User parameters.**

- **Right click on Thread Group-->Add two Sampler-->HTTP Request.**

Steps 1.20.2: Adding postprocessor

- **Right click on Test Plan-->Add-->Threads-->Thread Group.**

- **Right click on Thread Group-->Add -->Sampler-->HTTP Request1.**

- **Right click on PostProcessor Thread Group-->Add-->PostProcessor -->Result Status Action Handler.**

- **Right click on Thread Group-->Add -->Sampler-->Add another HTTP Request2 under Result Status Action Handler.**

- **Right click on PostProcessor -->Add --> Listener-->Run.**

23) Write a program to demonstrate XML and XML schema assertions in JMeter.
Adding XML assertion

The XML assertion is used to validate that the response follows a valid XML syntax. The steps to add XML Assertion in JMeter is given below:

- **Open the Apache JMeter.**
- **Click on the Test Plan.**
- **Rename the Test Plan.**
- **Right click on Test Plan --->Add--->Thread(users)--->ThreadGroup.**

- **Navigate through the given path Thread Group--->Name--->Comments---> Continue --->Thread Properties--->Number of Threads(users): ---> Ramp-Up Period (in seconds): ---> Loop Count --->Save.**

- **Right click on Thread Group --->Add---> Sampler---> HTTP Request.**

- **Go to HTTP Request---> Server Name or IP :---> Path :/---> Save.**

- **Right click on Thread Group --->Add--->Listeners---> View Results Tree---> Run.**

- **Right click on Thread Group --->Add--->Listeners---> View Results in Table---> Clear All---> Save---> Run.**

- **Right click on Thread Group --->Add--->Listeners---> Assertions Results---> Clear All---> Save---> Run.**

- **Right click on Thread Group --->Add--->Assertions ---> XML Assertions.**

- **Click on Save --->Clear All---> View Results in Table.**

Step 1.21.2: Adding XML schema assertion

The XML schema assertion is used to validate the response against a specified XML schema. Steps to add XML schema assertion in JMeter is shown below:

- Open the Apache JMeter.
- Click on the Test Plan.
- Rename the Test Plan.
- Right click on Test Plan --->Add--->Thread(users)--->Thread Group.
- Right click on Thread Group --->Add--->Assertions ---> XML Schema Assertions.

- Enter file name for XML schema---> Save.

- Click on Clear All---> Run ---> View Results in Table.

24) Write a program to demonstrate HTML assertions in JMeter.

Adding HTML assertions in JMeter

The HTML assertion is used to check the HTML syntax of the response.

- Open the Apache JMeter.
- Click on the Test Plan.
- Rename the Test Plan.
- Right click on Test Plan --->Add--->Thread(users)--->Thread Group.

- **Go to Thread Group--->Name--->Comments---> Continue --->Thread Properties--->Number of Threads(users): ---> Ramp-Up period (in seconds): ---> Loop Count --->Save.**

- **Right click on Thread Group --->Add---> Sampler---> HTTP Request.**

- **HTTP Request---> Server Name or IP :---> Path :/---> Save.**

- **Right click on Thread Group --->Add--->Listeners---> View Results Tree---> Run.**

- **Right click on Thread Group --->Add--->Listeners---> View Results in Table---> Clear All---> Save---> Run.**

- **Right click on Thread Group --->Add--->Listeners---> Assertions Results---> Clear All---> Save---> Run.**

- **Right click on Thread Group --->Add--->Assertions ---> HTML Assertions---> Error threshold---> Warning threshold--->Clear All---> Save.**

- **Go to Assertion Results---> Run. This gives an error.**

- **Go to HTML Assertions---> Error threshold: 0---> Warning threshold: 25--->Clear All---> Save---> Assertion Results---> Run.**

25) Write a program to demonstrate response assertions in Jmeter

Adding response assertion

Response assertion is used in test scripts to validate a pattern in the response body, header, code, message, etc. Steps to add response assertion in JMeter are shown below:

- **Open the Apache JMeter.**
- **Click on the Test Plan.**
- **Rename the Test Plan.**
- **Right click on Test Plan --->Add--->Thread(users)--->Thread Group.**
- **Go to Thread Group--->Name--->Comments---> Continue --->Thread Properties--->Number of Threads(users): ---> Ramp-Up Period (in seconds): ---> Loop Count --->Save.**
- **Right click on Thread Group --->Add---> Sampler---> HTTP Request.**
- **HTTP Request---> Server Name or IP :---> Path :/---> Save.**
- **Right click on Thread Group --->Add--->Listeners---> View Results Tree---> Run.**
- **Right click on Thread Group --->Add--->Listeners---> View Results in Table---> Clear All---> Save---> Run.**
- **Right click on Thread Group --->Add--->Listeners---> Assertions Results---> Clear All---> Save---> Run.**

- **Right click on Thread Group --->Add--->Assertions ---> Response Assertions---> Response Code---> Contains---> Add---> 201---> Save---> Clear All---> Run.**

- **Assertion Results---> Run. This will show an error.**

- **Right click on Thread Group --->Add--->Assertions ---> Response Assertions---> Response Code---> Contains---> Add---> 200---> Save---> Clear All---> Run.**

- **Right click on Thread Group --->Add--->Assertions ---> Response Assertions---> Response Message---> Contains---> Add---> OK---> Save---> Clear All---> Run ---> View Results in Table.**

26) Write a program to demonstrate XPath and Comparison assertions in JMeter.

Adding Xpath assertion

The Xpath assertion is used to validate the response using Xpath expressions. Steps to add Xpath assertion in JMeter are shown below:

- **Open the Apache JMeter.**
- **Click on the Test Plan.**
- **Rename the Test Plan.**
- **Right click on Test Plan --->Add--->Thread(users)--->Thread Group.**

- **Go to Thread Group--->Name--->Comments---> Continue --->Thread Properties--->Number of Threads(users): ---> Ramp-Up Period (in seconds): ---> Loop Count --->Save.**

- **Right click on Thread Group --->Add---> Sampler---> HTTP Request.**

- **HTTP Request---> Server Name or IP :---> Path :/---> Save.**

- **Right click on Thread Group --->Add--->Listeners---> View Results Tree---> Run.**

- **Right click on Thread Group --->Add--->Listeners---> View Results in Table---> Clear All---> Save---> Run.**

- Right click on Thread Group --->Add--->Listeners---> Assertions Results---> Clear All---> Save---> Run.

- Right click on Thread Group --->Add--->Assertions ---> Xpath Assertions---> Main sample only---> Quiet --->Xpath--->//*[@id=”search-2”]/format/input--->Save.

- Go to Assertion Result---> Run. This will show an error.

- Right click on Thread Group --->Add--->Assertions ---> Xpath Assertions---> Main sample only--->Use tidy(tolerant parser)---> Quiet --->Xpath--->//*[@id=”search-2”]/format/input--->Save.

- Click on Clear All---> Run--->View Results in Table.

Step 1.24.2: Adding compare assertion

The compare assertion checks the response content or confirms that response time of all samplers under the assertion scope are equal.

The request and the assertion should be on the same level in the test plan. Steps to add Xpath assertion is given below:

- Open the Apache JMeter.
- Click on the Test Plan.
- Rename the Test Plan.

- Right click on Test Plan --->Add--->Thread(users)--->Thread Group.

- Right click on Thread Group --->Add--->Assertions ---> Compare Assertions---> Compare Content---> Compare Time.

- Right click on View Results in Table---> Run.

27) Write a program to demonstrate random timers (Constant, Gaussian, and Poisson timers) in JMeter

: Demonstrating Gaussian Random timer

Gaussian Random timer delays each user request for a random amount of time.

- Open the Apache JMeter.

- Click on the Test Plan.

- Rename the Test Plan.

- Right click on Test Plan --->Add--->Thread(users)--->Thread Group.

- Go to Thread Group--->Name--->Comments---> Continue --->Thread Properties--->Number of Threads(users): ---> Ramp-Up Period (in seconds): ---> Loop Count --->Save.

- Right click on Thread Group --->Add---> Sampler---> HTTP Request.

- **Go to HTTP Request---> Server Name or IP :---> Path :/---> Save.**

- **Right click on Thread Group --->Add--->Listeners---> View Results Tree---> Run.**

- **Right click on Thread Group --->Add--->Listeners---> View Results in Table---> Clear All---> Save---> Run.**

- **Right click on Thread Group --->Add--->Listeners---> Assertions Results---> Clear All---> Save---> Run.**

- **Right click on Thread Group---> Add---> Timer---> Gaussian Random Timer.**

- **Go to Gaussian Random Timer--->Deviation in ms---> Constant Delay Offset---> Save.**

- **Click on Clear All---> Run---> View Results in Table.**

Step 1.25.2: Demonstrating Poisson Random timer

To pause each and every thread request for a random amount of time use Poisson Random timer.

- **Right click on Thread Group---> Add---> Timer---> Poisson Random Timer--->Lambda in ms---> Constant Delay Offset---> Save.**

- **Go to Clear all---> Run---> View Results in Table.**

Step 1.25.3: Demonstrating Uniform Random timer

Uniform Random timer delays each user request for a random amount of time.

- **Right click on Thread Group---> Add---> Timer---> Uniform Random Timer---Random Delay Maximum in ms---> Constant Delay offset---> Save.**

- **Click on Clear all---> Run---> View Results in Table.**

Step 1.25.4: Demonstrating Constant timer

Constant timer delays each user request for the same amount of time.

- **Right click on Thread Group---> Add---> Timer---> Constant Timer---Thread Delay in ms---> Save.**

- **Click on Clear all---> Run---> View Results in Table.**

28) Write a program to demonstrate how a timer is synchronized in JMeter.

- **Open Apache JMeter.**

- **Click on the Test Plan.**

- **Rename the Test Plan.**

- **Right click on Test Plan --->Add--->Thread(users)--->Thread Group.**

- **Thread Group--->Name--->Comments---> Continue --->Thread Properties--->Number of Threads(users): ---> Ramp-Up Period (in seconds): ---> Loop Count --->Save.**

**●	Right click on Thread Group --->Add---> Sampler---> HTTP Request.**

**●	Click on HTTP Request---> Server Name or IP :---> Path :/---> Save.**

**●	Right click on Thread Group --->Add--->Listeners---> View Results Tree---> Run.**

**●	Right click on Thread Group --->Add--->Listeners---> View Results in Table---> Clear All---> Save---> Run.**

**●	Right click on Thread Group --->Add--->Listeners---> Assertions Results---> Clear All---> Save---> Run.**

**●	Right click on Thread Group---> Add---> Timer---> Synchronizing timer--->Number of Simulated Users to Group by ---> Timeout in ms--->Save.**

**●	Click on Clear all---> Run---> View Results in Table.**

29) Write a program to demonstrate random variables and counters in JMeter.

Generating values by random variable element

**●	Sometimes we need to generate random values in our JMeter scripts and then work with them in our requests. For example, instances where the URL requires an auto-generated and unique value as a parameter.**

**●	There are many ways to do this in JMeter and we will look at some of the most common ones.**

- **Adding random variable element:**

1. **Open JMeter.**
2. **Right click on the Test Plan.**
3. **Right click on Thread Group.**
4. **Click on Add -> Config Element -> Random Variable.**

5. **The sampler is as shown below and you need to fill in the required fields:**

I. Variable Name: The name used to invoke the variable (Ex: random)

II. Output Format: The specified format for the variable (Ex: user_00000)

III. Minimum and Maximum Value: Range of the variable (Ex: 10000 to 99999)

IV. Per Thread: If you set it to True, the value will be shared by the threads. This means that there will be threads with the same value. If you want to always generate a different value, you have to set it to False.

- **Adding HTTP request:**

1. **Right click on Thread Group.**
2. **Click on Add -> Sampler -> HTTP Request.**

3. **Rename HTTP request as “RandomValue_${random}” where random is the variable name specified in random variable element.**
4. **${variable_name} is used to get the value of the variable.**
5. **Specify server name and path.**

- **Adding View Results Tree:**

1. Right click on Thread Group.

2. Click on Add -> Listener -> View Results Tree.

3. Open Thread Group.

4. Set the Loop Count to 10.

5. Run the Thread Group and open View Results Tree to see the output.

6. The mentioned 10 HTTP Request runs with different random values with specified format "user_00000" Ex: RandomValue_user_40380.

Step 1.27.2: Generating random values by _time()

- **Adding users in thread group:**

1. Right click on Test Plan.

2. Click on Add -> Threads(users) -> Thread Group.

1. Set the Number of Threads to 3.

- **Adding View Results Tree:**

1. Right click on Thread Group.

2. Click on Add -> Listener -> View Results Tree.

- **Adding HTTP Request:**

1. Right click on Thread Group.

2. Click on Add -> Sampler -> HTTP Request.

3. Rename HTTP request to Http_Request_${__time()} , where "__time()" gives the current time in seconds.

4. Specify the server name and path in the HTTP request.

5. Run the Thread Group and Open the View Results Tree to see the output.

6. Three HTTP Requests run with different values.

Step 1.27.3: Adding counter element

Let's imagine a scenario in which you need to create five entities in a loop using the HTTP Request sampler and each entity name has to be unique. In this case, the better way is to use a Counter is by:

● Adding users in thread group:

1. Open JMeter.

2. Right click on the Test Plan.

3. Click on Add -> Threads(users) -> Thread Group.

4. Set the Loop Count to 10.

● Adding View Results Tree:

1. Right click on Thread Group.

2. Click on Add -> Listener -> View Results Tree.

● Adding Counter Element:

1. Right click on Thread Group.

2. Click on Add -> Config Element -> Counter.

3. Enter the required fields in the counter.

i. Starting value: Initial counter value, Example: 1

ii. Increment: This value will be added to the current counter value once the counter is encountered (Ex: 1).

iii. Maximum value: When the current counter value exceeds "maximum" value, counter starts over. If it is left blank, the counter value will increase infinitely. Ex: if counter maximum value is five and number of threads are ten, then after counter value reaches five the counter value starts from 1.

iv. Number format: The number format will be Ex: 000.

v. Reference name: The name used to invoke the variable (Ex: "C")

● Adding HTTP Request:

1. Right click on Thread Group.

2. Click on Add -> Sampler -> HTTP Request.

3. Rename the HTTP Request to "counter_${C}" , where "C" is the reference name of the counter.

4. Enter the server name and path.

5. Run the Thread Group and open View Results Tree to see the output.

Step 1.27.4: Pushing the code to GitHub repositories

30) Write a program to demonstrate configuration elements for HTTP, FTP, Java, JDBC, and UDV in JMeter.

Using HTTP authentication manager

● Let's take an example of "https://httpbin.org/basic-auth/user/passwd".

● When you click on the https://httpbin.org/basic-auth/:user/:passwd link, your browser should prompt you to enter a username and password.

- **Provide the default username as user and the default password as passwd. The server should respond with the following json:**

- **For wrong username and password, the server rejects access to the protected resource with an HTTP 401: Unauthorized error.**

- **Adding users in thread group:**

1. **Right click on Test Plan.**
2. **Click on Add -> Threads(users) -> Thread Group.**
3. **Set Number of Threads to 1.**
4. **Set the Ramp-Up period to 1 Second.**
5. **Set Loop Count to 1.**

- **Adding HTTP Authorization Manager:**

1. **Right click on Thread Group.**
2. **Click on Add -> Config Element -> HTTP Authorization Manager.**
3. **Enter the required fields in HTTP Authorization Manager.**

I. Base URL: https://httpbin.org/basic-auth/user/passwd

II. Username: user

III. Password: passwd

IV. Mechanism: BASIC_DIGEST

- **Adding HTTP Request:**

1. **Right click on Thread Group.**
2. **Click on Add -> Sampler -> HTTP Request.**
3. **Enter protocol: https.**
4. **Enter server name: httpbin.org.**
5. **Enter path: /basic-auth/user/passwd.**

- **Adding View Results Tree:**

1. **Right click on Thread Group.**
2. **Click on Add -> Listener -> View Results Tree.**

3. **Run the Thread Group and open the View Results Tree to see the output.**

I. Enter valid username: user and password: passwd.

II. The Response code: 200

III. Response Message: OK

IV. Response Data:

```
                        {
                                    "authenticated": true,
                                    "user": "user"
}
```

For invalid username and password or By Disabling HTTP Authorization Manager the output will be:

I. Response Code: 401

II. Response Message: UNAUTHORIZED

Step 1.29.2: Using HTTP Cookie Manager

In a stateless internet, many sites and applications use cookies to retain a handle between sessions or to keep some state on the client side. If you are planning to use JMeter to test such web applications, then you will want to use HTTP Cookie Manager.

- Adding users in thread group:

1. Right click on Test Plan.
2. Click on Add -> Threads -> Thread Group.

3. Set the number of threads to 1.
4. Set the ramp-up period to 1.

5. Set loop count to 1.

- **Adding HTTP request:**

1. Right click on Thread Group.

2. Click on Add -> Sampler -> HTTP Request.

3. Rename HTTP request to “HTTP Post”.

4. Set the protocol to “https”.

5. Set the server name to “www.youtube.com”.

6. Set method to “POST”.

7. Set path to “/results?search_query”.

8. Add the parameters:

I. Name: search_query

II. Value: java

III. Check URL ENCODE?

IV. Check INCLUDE EQUALS?

- **Adding HTTP Request:**

1. Right click on Thread Group.

2. Click on Add -> Sampler -> HTTP Request.

3. Rename HTTP request to “HTTP Post 1”.

4. Set the protocol to “https”.

5. Set the server name to “www.youtube.com”.

6. **Set method to “POST”.**

7. **Set path to “/results?search_query”.**

8. **Add the parameters:**

I. **Name: search_query**

II. **Value: jmeter**

III. **Check URL ENCODE?**

IV. **Check INCLUDE EQUALS?**

- **Adding HTTP Cookie Manager:**

1. **Right click on Thread Group.**

2. **Click on Add -> Config Element -> HTTP Cookie Manager.**

3. **Check the “Clear Cookies each iteration?”.**

4. **Select the cookie policy “rfc2109”.**

- **Adding View Results Tree:**

1. **Right click on Thread Group.**

2. **Click on Add -> Listener -> View Results Tree.**

3. Run the Thread Group and open the View Results Tree to see the result.

4. As you can see, HTTP Post Request Cookie is set as shown below:

Note that in the additional request HTTP Post 1, the cookie value is the same as when it was received directly from the server.

- Disable HTTP cookie manager:

1. Right click on HTTP cookie manager.

2. Click on Disable.

3. Run the Thread Group and open View Results Tree to see the result.

Note that for both the post requests, different cookie values are set.

Step 1.29.3: Reading values from CSV file using CSV Data Set Config

- JMeter is an open source load testing tool that has an element that allows you to use external data sets in a CSV format. This element is called the “CSV Data Set Config”. The CSV Data Set Config is used to read lines from a file and to split them into variables.

- **Adding users in thread group:**

1. **Right click on Test Plan.**
2. **Click on Add -> Threads -> Thread Group.**

3. **Set Number of threads to 1.**
4. **Set Ramp-Up Period to 1.**
5. **Set Loop Count to 3.**

6. **Imagine we have a list of random variables that are already generated and saved as "book2.csv".**

- **Now, we want to read this list in our request. So, we have to add a CSV data set Config as we show below:**

- **Adding CSV Data Set Config:**

1. **Right click on Thread Group.**
2. **Click on Add -> Config Element -> CSV Data Set Config.**

3. Set the CSV Data Set Config as follows:

i. Filename: If your file is in the /bin directory, this can just be the filename. If it's somewhere else, use the full path to the file.

ii. Variable Names: Comma-delimited list of names should be equal number of names than columns in CSV file, example: random, id.

iii. Ignore First Line: Set as False by default. Used whenever you want to ignore the first line as value (because it may contain the column names).

iv. Delimiter: A comma is the default delimiter.

v. Allow Quoted Data: Set to False by default, set to True if you have double quotes in CSV columns.

vi. Recycle on EOF: True by default, JMeter restarts from the beginning when the End Of File is reached.

vii. Stop Thread on EOF: Stops the current thread gathering the value if EOF is reached

viii. Sharing Mode: Defines how values are distributed among concurrent threads

● Adding HTTP Request:

1. Right click on Thread Group.

2. Click on Add -> Sampler -> HTTP Request.

3. Rename HTTP Request to "DataFromCSV_ColumnOne_${random}_ColumnSecond_${id}" , where random, id are variable names from CSV Data Set Config.

4. Give protocol as "https".

5. Give server name as "www.wikipedia.org".

6. Give path "/".

- **Adding View Results Tree:**

1. **Right click on Thread Group.**
2. **Click on Add -> Listener -> View Results Tree.**
3. **Run Thread Group and open View Results Tree to see the output.**

31) Write a program to demonstrate Http Authentication Manager, cookies, and CSV Data Set Config in JMeter.

Using HTTP authentication manager

- **Let's take an example of "https://httpbin.org/basic-auth/user/passwd".**
- **When you click on the https://httpbin.org/basic-auth/:user/:passwd link, your browser should prompt you to enter a username and password.**
- **Provide the default username as user and the default password as passwd. The server should respond with the following json:**
- **For wrong username and password, the server rejects access to the protected resource with an HTTP 401: Unauthorized error.**
- **Adding users in thread group:**

1. **Right click on Test Plan.**
2. **Click on Add -> Threads(users) -> Thread Group.**

3. Set Number of Threads to 1.

4. Set the Ramp-Up period to 1 Second.

5. Set Loop Count to 1.

- Adding HTTP Authorization Manager:

1. Right click on Thread Group.

2. Click on Add -> Config Element -> HTTP Authorization Manager.

3. Enter the required fields in HTTP Authorization Manager.

I. Base URL: https://httpbin.org/basic-auth/user/passwd

II. Username: user

III. Password: passwd

IV. Mechanism: BASIC_DIGEST

- Adding HTTP Request:

1. Right click on Thread Group.

2. Click on Add -> Sampler -> HTTP Request.

3. Enter protocol: https.

4. Enter server name: httpbin.org.

5. Enter path: /basic-auth/user/passwd.

- **Adding View Results Tree:**

1. Right click on Thread Group.

2. Click on Add -> Listener -> View Results Tree.

3. Run the Thread Group and open the View Results Tree to see the output.

I. Enter valid username: user and password: passwd.

II. The Response code: 200

III. Response Message: OK

IV. Response Data:

```
{
    "authenticated": true,
    "user": "user"
}
```

For invalid username and password or By Disabling HTTP Authorization Manager the output will be:

I. Response Code: 401

II. Response Message: UNAUTHORIZED

Step 1.29.2: Using HTTP Cookie Manager

In a stateless internet, many sites and applications use cookies to retain a handle between sessions or to keep some state on the client side. If you are planning to use JMeter to test such web applications, then you will want to use HTTP Cookie Manager.

- **Adding users in thread group:**

1. **Right click on Test Plan.**
2. **Click on Add -> Threads -> Thread Group.**
3. **Set the number of threads to 1.**
4. **Set the ramp-up period to 1.**
5. **Set loop count to 1.**

- **Adding HTTP request:**

1. **Right click on Thread Group.**
2. **Click on Add -> Sampler -> HTTP Request.**
3. **Rename HTTP request to “HTTP Post”.**
4. **Set the protocol to “https”.**
5. **Set the server name to “www.youtube.com”.**
6. **Set method to “POST”.**

7. **Set path to “/results?search_query”.**

8. **Add the parameters:**

I. **Name: search_query**

II. **Value: java**

III. **Check URL ENCODE?**

IV. **Check INCLUDE EQUALS?**

- **Adding HTTP Request:**

1. **Right click on Thread Group.**

2. **Click on Add -> Sampler -> HTTP Request.**

3. **Rename HTTP request to “HTTP Post 1”.**

4. **Set the protocol to “https”.**

5. **Set the server name to “www.youtube.com”.**

6. **Set method to “POST”.**

7. **Set path to “/results?search_query”.**

8. **Add the parameters:**

I. **Name: search_query**

II. **Value: jmeter**

III. **Check URL ENCODE?**

IV. **Check INCLUDE EQUALS?**

- **Adding HTTP Cookie Manager:**

1. **Right click on Thread Group.**
2. **Click on Add -> Config Element -> HTTP Cookie Manager.**

3. **Check the “Clear Cookies each iteration?”.**
4. **Select the cookie policy “rfc2109”.**

- **Adding View Results Tree:**

1. **Right click on Thread Group.**
2. **Click on Add -> Listener -> View Results Tree.**

3. **Run the Thread Group and open the View Results Tree to see the result.**
4. **As you can see, HTTP Post Request Cookie is set as shown below:**

Note that in the additional request HTTP Post 1, the cookie value is the same as when it was received directly from the server.

- **Disable HTTP cookie manager:**

1. **Right click on HTTP cookie manager.**
2. **Click on Disable.**
3. **Run the Thread Group and open View Results Tree to see the result.**

Note that for both the post requests, different cookie values are set.

Step 1.29.3: Reading values from CSV file using CSV Data Set Config

- **JMeter is an open source load testing tool that has an element that allows you to use external data sets in a CSV format. This element is called the "CSV Data Set Config". The CSV Data Set Config is used to read lines from a file and to split them into variables.**

- **Adding users in thread group:**

1. **Right click on Test Plan.**
2. **Click on Add -> Threads -> Thread Group.**

3. **Set Number of threads to 1.**
4. **Set Ramp-Up Period to 1.**
5. **Set Loop Count to 3.**

6. Imagine we have a list of random variables that are already generated and saved as "book2.csv".

- Now, we want to read this list in our request. So, we have to add a CSV data set Config as we show below:

- Adding CSV Data Set Config:

1. Right click on Thread Group.

2. Click on Add -> Config Element -> CSV Data Set Config.

3. Set the CSV Data Set Config as follows:

i. Filename: If your file is in the /bin directory, this can just be the filename. If it's somewhere else, use the full path to the file.

ii. Variable Names: Comma-delimited list of names should be equal number of names than columns in CSV file, example: random, id.

iii. Ignore First Line: Set as False by default. Used whenever you want to ignore the first line as value (because it may contain the column names).

iv. Delimiter: A comma is the default delimiter.

v. Allow Quoted Data: Set to False by default, set to True if you have double quotes in CSV columns.

vi. Recycle on EOF: True by default, JMeter restarts from the beginning when the End Of File is reached.

vii. Stop Thread on EOF: Stops the current thread gathering the value if EOF is reached

viii. Sharing Mode: Defines how values are distributed among concurrent threads

- Adding HTTP Request:

1. Right click on Thread Group.

2. Click on Add -> Sampler -> HTTP Request.

3. Rename HTTP Request to “DataFromCSV_ColumnOne_${random}_ColumnSecond_${id}” , where random, id are variable names from CSV Data Set Config.

4. Give protocol as “https”.

5. Give server name as “www.wikipedia.org”.

6. Give path “/”.

- Adding View Results Tree:

1. Right click on Thread Group.

2. Click on Add -> Listener -> View Results Tree.

3. Run Thread Group and open View Results Tree to see the output.

Note: We can see the values defined in the CSV file.

32) Demonstrate processors, assertions, and timers in JMeter.

: Problem statement for Processors, Assertions, and Timers

- **Objective: Create Processors, Assertions, Timers in Jmeter for a particular domain.**
- **Steps involved:**

1) Set up JMeter.

2) Create a test plan.

3) Add thread group.

4) Add HTTP request.

5) Record the scripts.

6) Add Listener and view the results.

Step 1.3.2: Solution for the problem statement

1) Adding Preprocessor:

- **Right click on test plan-->Add-->Threads-->Thread Group.**

- **Right click on Thread-->Add-->preprocessor -->User parameters.**

- **Right click on Thread Group-->Add Two-->Sampler-->HTTP Request-->consider facebook.com domain.**

2) Adding Post Processor:

- **Right click on test plan-->Add-->Threads-->Thread Group.**

- **Right click on Thread Group-->Add -->Sampler-->HTTP Request1.**

- Right click on post processor Thread Group-->Add-->post processor -->Result Status Action Handler.

- Right click on Thread Group-->Add -->Sampler-->Another HTTP Request2 under Result Status Action Handler.

- Right click on postprocessor -->Add --> Listener-->Run result.

3) Response Assertion:

The Response assertion is used in test scripts to validate a pattern in the response body, header, code, message, etc.

- **Open the Apache JMeter.**
- **Click on the Test plan.**
- **Rename the Test plan.**
- **Right click on Test plan --->Add--->Thread(users)--->Thread Group.**

- **Navigate through the path: Thread Group--->Name--->Comments---> Continue --->Thread properties--->Number of Threads(users): ---> Ramp-up period (in seconds): ---> Loop Count --->Save.**

- **Right click on Thread group --->Add---> Sampler---> HTTP Request.**

- **Navigate through the given path: HTTP Request---> Server Name or IP :---> Path :/---> Save.**

- **Right click on Thread Group --->Add--->Listeners---> View Results Tree---> Run.**

- **Right click on Thread Group --->Add--->Listeners---> View Results in Table---> Clear All---> Save---> Run.**

- **Right click on Thread Group --->Add--->Listeners---> Assertions Results---> Clear All---> Save---> Run.**

- **Right click on Thread Group --->Add--->Assertions ---> Response Assertions---> Response Code---> Contains---> Add---> 201---> Save---> Clear All---> Run.**

- **Navigate through the given path: Assertion Result---> Run. It will show error**

- **Right click on Thread Group --->Add--->Assertions ---> Response Assertions---> Response Code---> Contains---> Add---> 200---> Save---> Clear All---> Run.**

- **Right click on Thread Group --->Add--->Assertions ---> Response Assertions---> Response Message---> Contains---> Add---> OK---> Save---> Clear All---> Run ---> View Results in Table.**

4) Constant Timer:

- Constant timer delays each user request for the same amount of time.

- Right click on Thread Group---> Add---> Timer---> Constant Timer---Thread delay in ms---> Save.

- Navigate through the given path: Clear all---> Run---> View Result in Table.

33) Write a program to demonstrate distributed config on JMeter.

Explaining prerequisites for JMeter distributed

- Master and slave machines have JMeter installed with the same versions.

- Master and slave machines must have the same versions of Java.

- Master and slave should connect to a common subnet.

- Open JMeter properties file, ind remote host and edit remote_host to 192.168.95.1 as shown below:

- Run create-rmi-keystore.bat and enter these fields, name:rmi, pwd : changeit.

- Rmi-keystore .jks will be generated automatically.

- Run jmeter-server.bat.

Step 1.30.2: Performing distributed testing with JMeter

- Right click on Test Plan.

- Click on Thread(Users)->Thread Group.

- Right click on Thread Group->Add->Samplers->HTTP Request-> Enter server name "www.google.com".

- Right click on Thread Group->Add->Listener->View Results Tree.

- Run Remote Server.

- The screenshot below shows the final result in the tree.

- The screenshot below shows the final result in the server.

Step 1.30.3: Pushing the code to your GitHub repositories