

JMeter



Dr. D. P. Mohapatra

Professor

**Department of Computer Science & Engineering
National Institute of Technology, Rourkela-769008
durga@nitrkl.ac.in**

Outline

- History of JMeter
- JMeter Features
- Basic Terminologies
- Samplers
- Timers
- Logic Controllers
- Assertions
- Conclusion

History of JMeter

- It is a product of Apache
- [Stefano Mazzocchi](#) of the Apache Software Foundation (ASF) was the original developer of JMeter
 - He developed JMeter to test the performance of Apache JServ (now called [Apache Tomcat](#) project)
- ASF later redesigned JMeter to enhance the GUI
- Latest version of JMeter is 2.12

JMeter Features

- JMeter is an open source software tool and is platform-independent
- A tool to perform *load testing* in web server
 - Load testing measure performance of a system in terms of frequency of failure under heavy load
- Used to test heavy loads on a web server or group of web servers
- Used to perform load testing on a variety of web services such as HTTP, FTP, JDBC enabled database, SMTP, POP, IMAP, MongoDB and many more

JMeter Features cont..

- Provides different **Samplers** to test different web services
- Provides **Listeners** to analyze the results that are received after a test
- Provides in-built set of **Timers** which can create time gaps between tests or between firing requests by the users
- Provides a number of **Logic Controllers** to control the test script
- Provides error handling facilities using **Assertions**
- Supports a set of **templates** for different kinds of tests
- Supports a number of **plug-ins** which can extend its features as per dynamic updates

Basic Terminologies

- **Test Plan:** Used to create a JMeter script which can run tests
- **Threads:** A thread simulate a user. In one test plan, a user can create many threads as per the requirements for load testing
- **ThreadGroup:** It contains a group of threads or virtual users, which are created to perform different activities at a time
- **Sampler:** Used to make a request and a user can create different types of requests such as HTTP, JDBC, FTP, and SMTP etc.
- **Listeners:** These are used for results analysis
 - JMeter supports a variety of Listeners such as View Results in Table, View Results Tree etc.
- **Timers:** Used to add some delay to a sampler

Basic Terminologies cont...

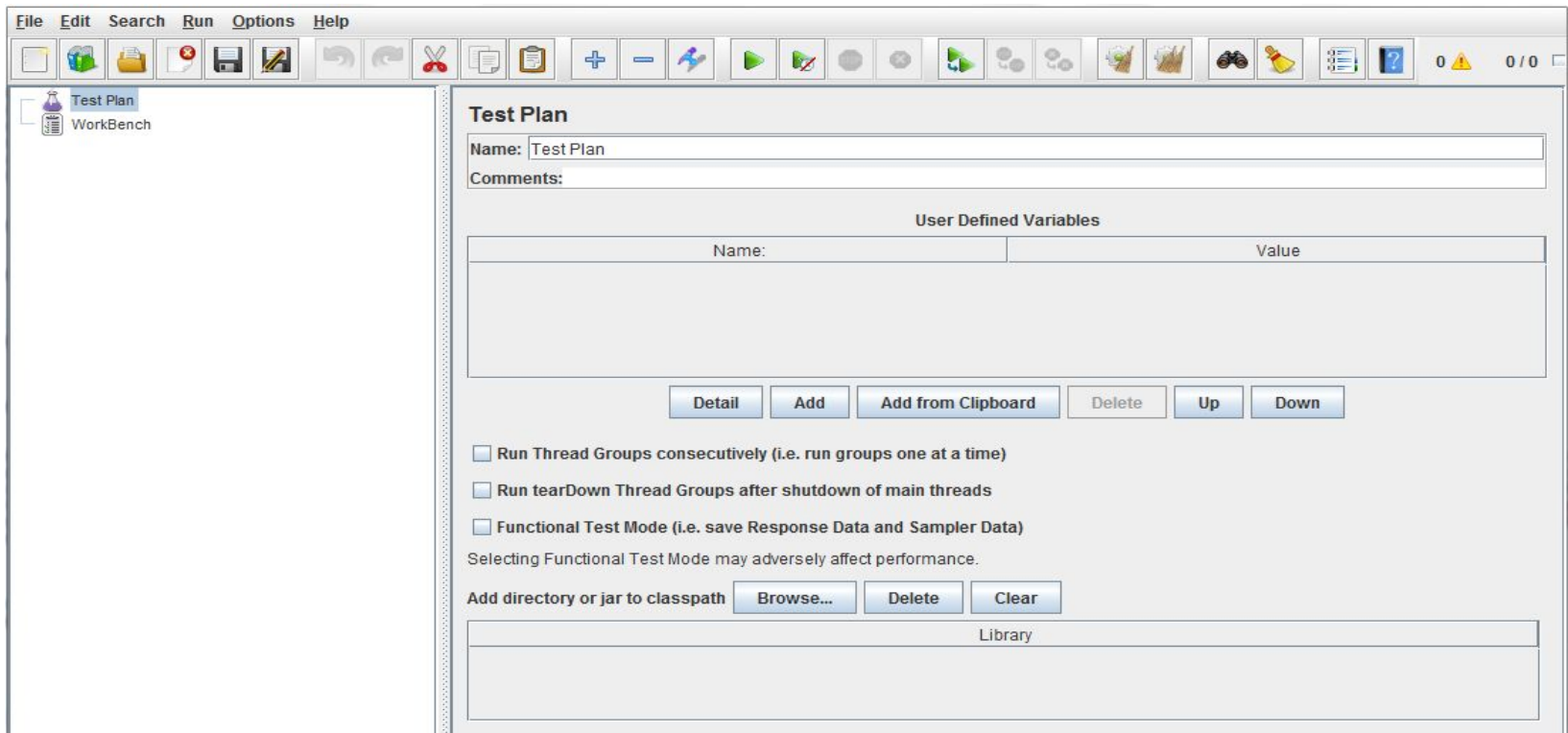
- **Controllers:** These are used to process the ordering of samplers and these are the logic expressions which perform an activity upon fulfillment of a certain condition
- **Assertions:** These act as checkpoints and are used for error handling
- **Config Elements:** Used to enhance a test script by adding extra functionality in combination with a sampler
- **Workbench:** This is the temporary workspace and is used to store the scripts

Basic Terminologies cont...

- **Pre-Processor Elements:** Used to modify a request that needs to be sent to the server
- **Post-Processor Elements:** Used to parse the received responses and extract a particular value from the parsed output
- **HTTP Proxy Server:** This is a service used by JMeter which acts as a proxy server in the network and is used to record user activities
- **HTTP Cookie Manager:** This feature of JMeter can simulate a real cookie. The Cookie Manager stores the cookies and while replaying it uses the data stored in the cookies

JMeter GUI

- To run JMeter, open the file */bin/Jmeter.bat* in Windows



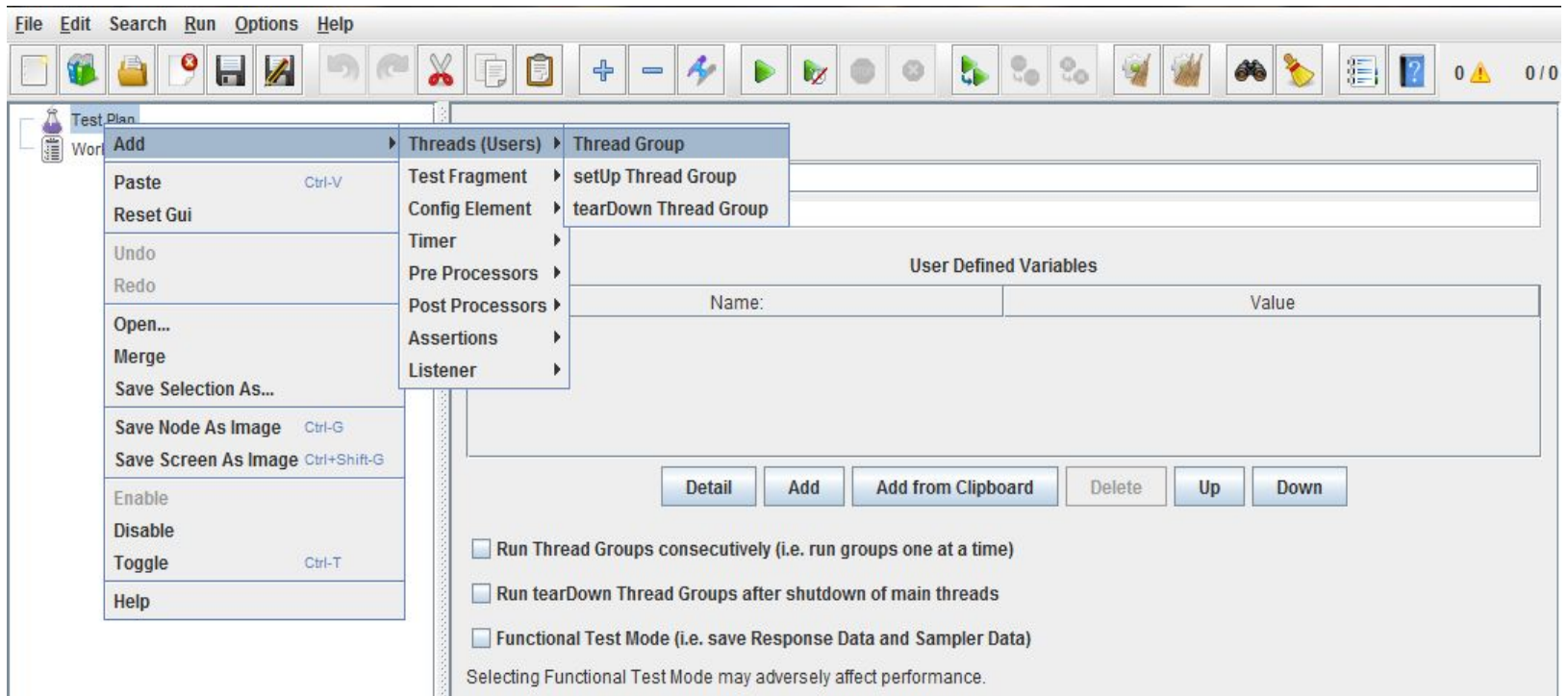
A Screenshot of JMeter GUI Version 2.9

Running JMeter in Server Mode

- To start the JMeter in server mode, open the file [/bin/jmeter-server.bat](#)
- In server mode, JMeter distributes its test parts to its clients that is it performs load testing by providing heavy loads using the client nodes

Running a Basic Test Script

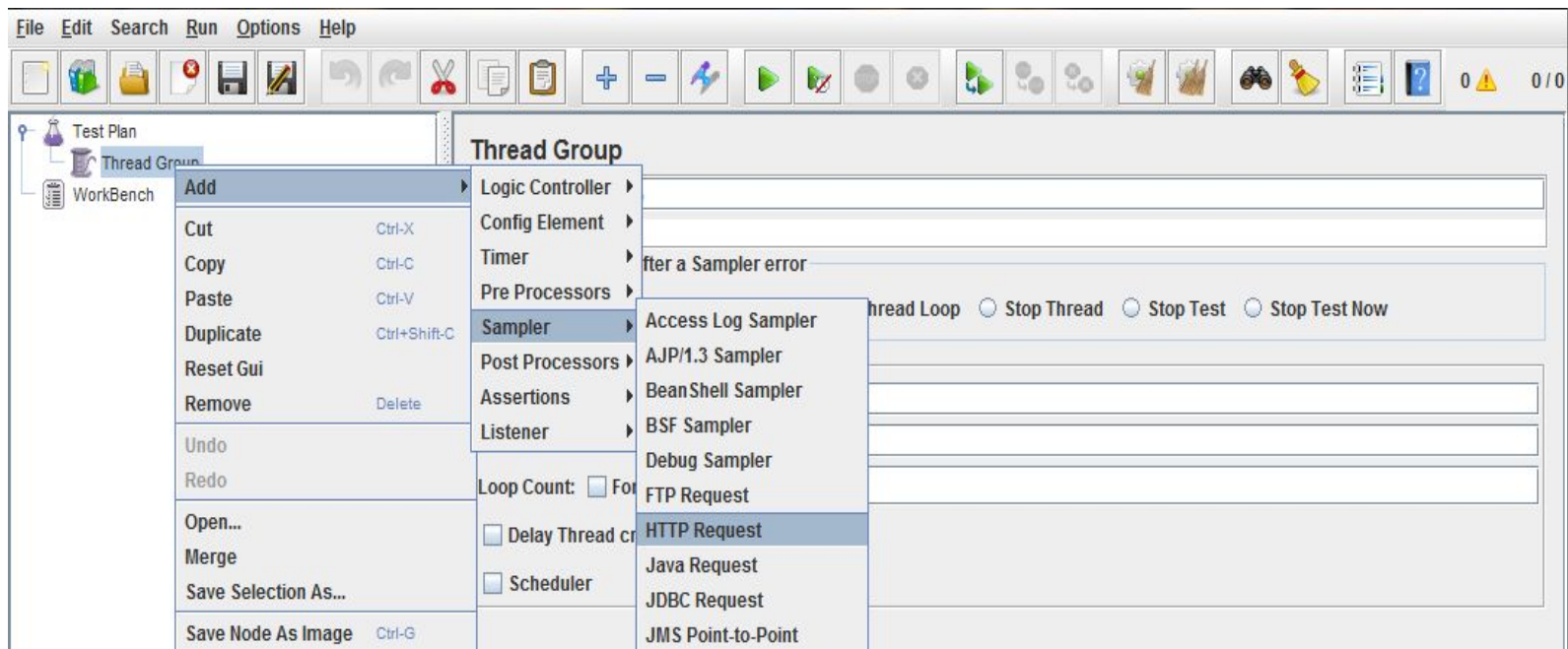
- Step1: Right click on Test Plan and go to *Add>Threads(Users)>Thread Group*



Creating a Thread Group

Running a Basic Test Script cont..

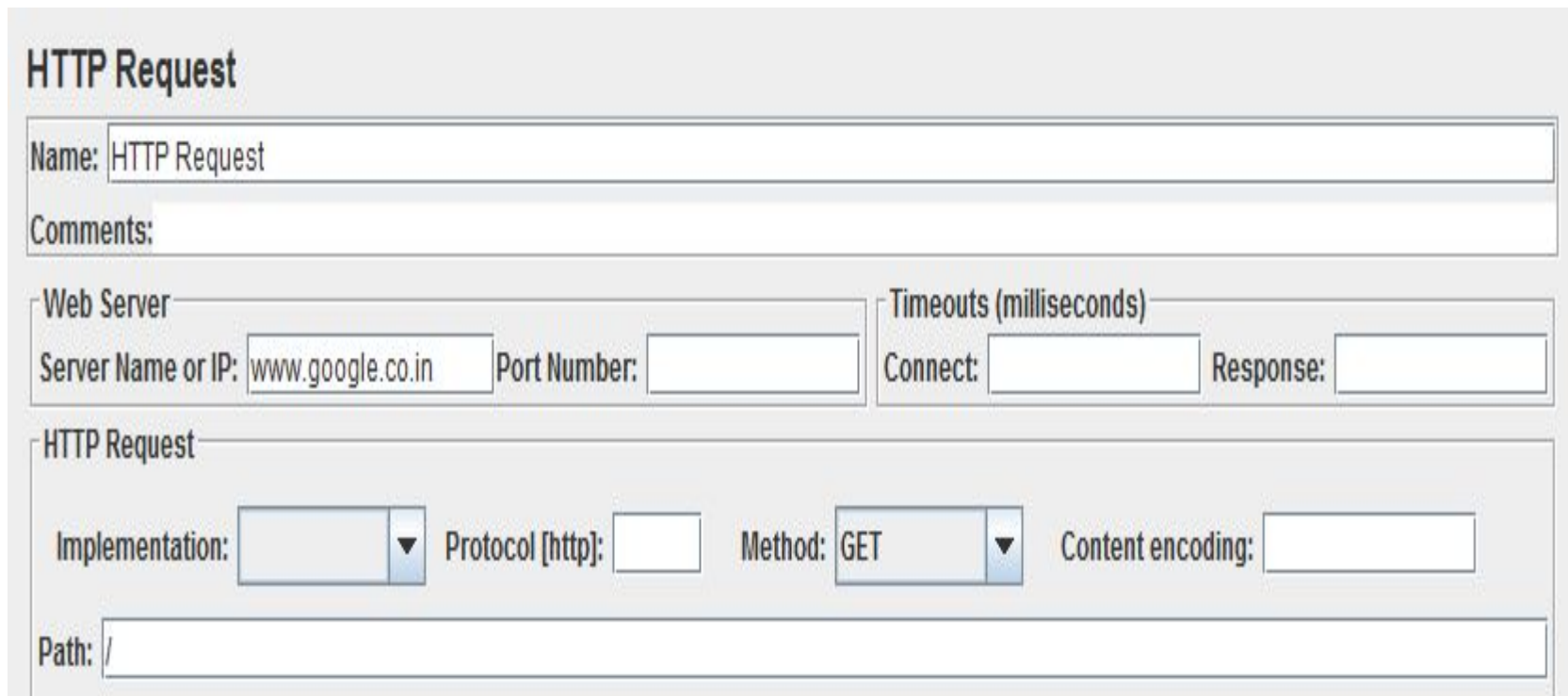
- **Step 2:** After creating Thread Group, right click on *Thread group* on the test plan pane and select *Add>Sampler>HTTP Request*



Adding a Sampler

Running a Basic Test Script cont..

- **Step 3:** Fill the server name field by *www.google.co.in* and path by /



The image shows a screenshot of the 'HTTP Request' form in JMeter. The form is titled 'HTTP Request' and contains several fields for configuring an HTTP request. The 'Name' field is set to 'HTTP Request'. The 'Comments' field is empty. The 'Web Server' section includes 'Server Name or IP' set to 'www.google.co.in', 'Port Number' (empty), 'Timeouts (milliseconds)' (empty), 'Connect' (empty), and 'Response' (empty). The 'HTTP Request' section includes 'Implementation' (empty), 'Protocol [http]' (empty), 'Method' set to 'GET', 'Content encoding' (empty), and 'Path' set to '/'. The form is styled with a light gray background and white input fields.

HTTP Request

Name: HTTP Request

Comments:

Web Server

Server Name or IP: www.google.co.in Port Number: Timeouts (milliseconds) Connect: Response:

HTTP Request

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

Path: /

HTTP Request Sampler Form

Running a Basic Test Script cont..

- **Step 4:** To collect the results, add a listener by right click on Thread Group, select *Add>Listener>View Results in Table*

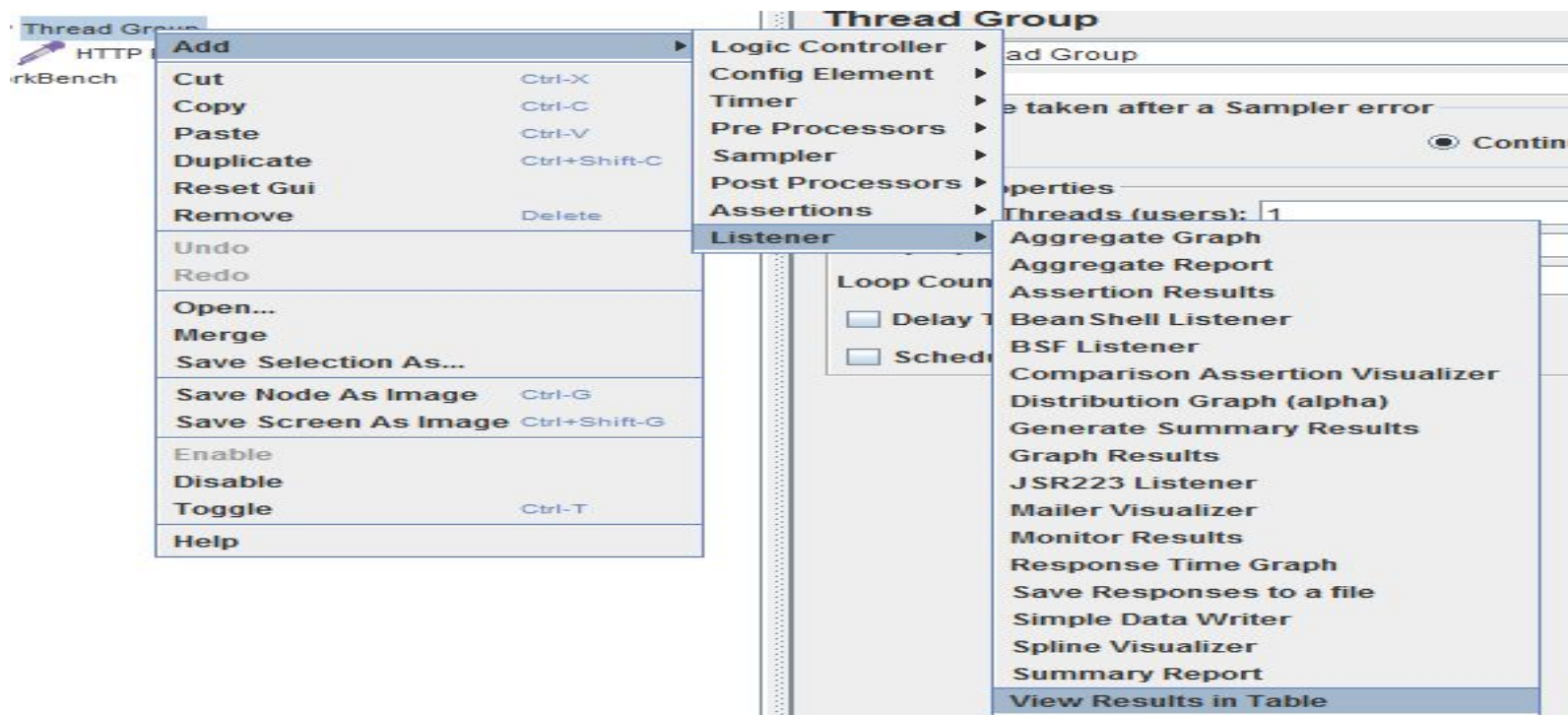
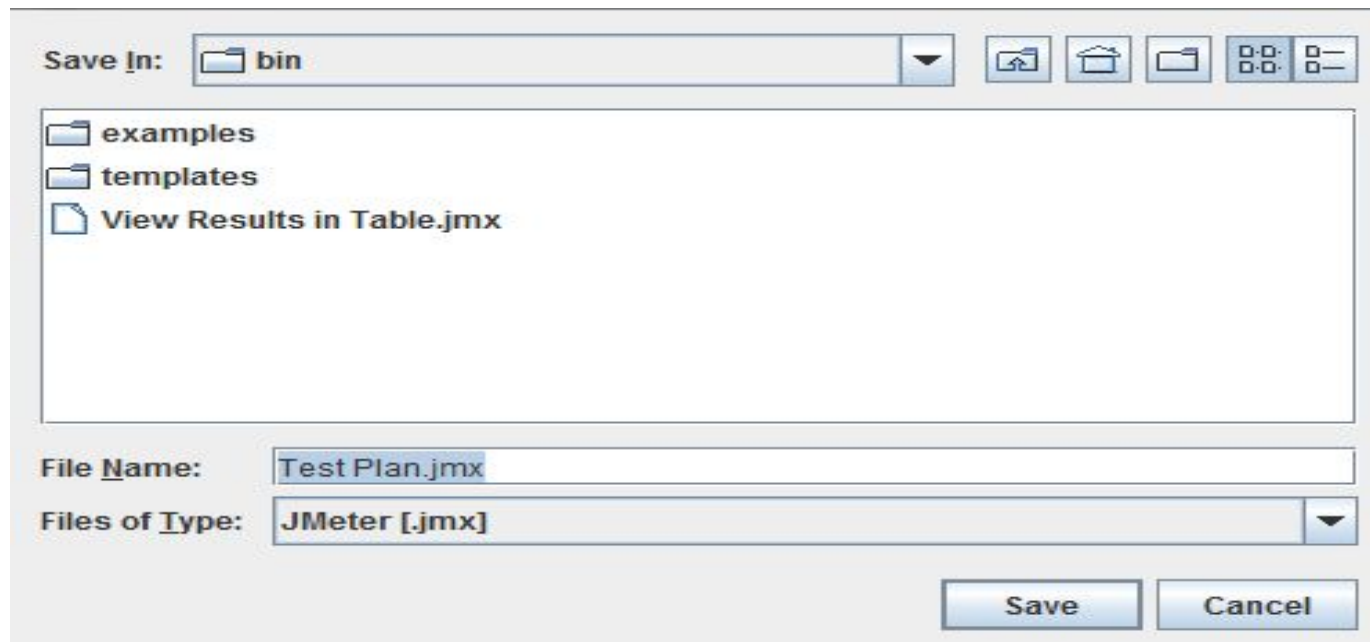


Table Listener Form

Running a Basic Test Script

- **Step 5:** Save the test script by pressing *ctrl + S* or go to File menu and click on Save and fill the Save Dialog box



Save Dialog Box

Running a Basic Test Script

- **Step 6:** Now you can see the results in the *Table Listener* by clicking on *View Results in Table* under the test script tree


View Results in Table

Name:

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency
1	11:41:17.146	Thread Group 1-1	HTTP Request	1329		20218	1040

HTTP Sampler Form

Samplers in JMeter

JMeter provides a number of samplers to test different web services.

- **FTP Request Sampler:** Used to test File Transfer Protocol (FTP) service. FTP services provide data for download in a secured manner
- **SMTP Sampler:** Used to test the E-Mail services
- **Mail Reader Sampler:** Used to read mail from the mail servers. It uses different protocols for reading mails which includes POP3, IMAP, IMAPS, etc.
- **Debug Sampler:** This is used to debug JMeter itself. When a script behaves abnormally, the Debug Sampler is used to debug it

Samplers in JMeter cont..

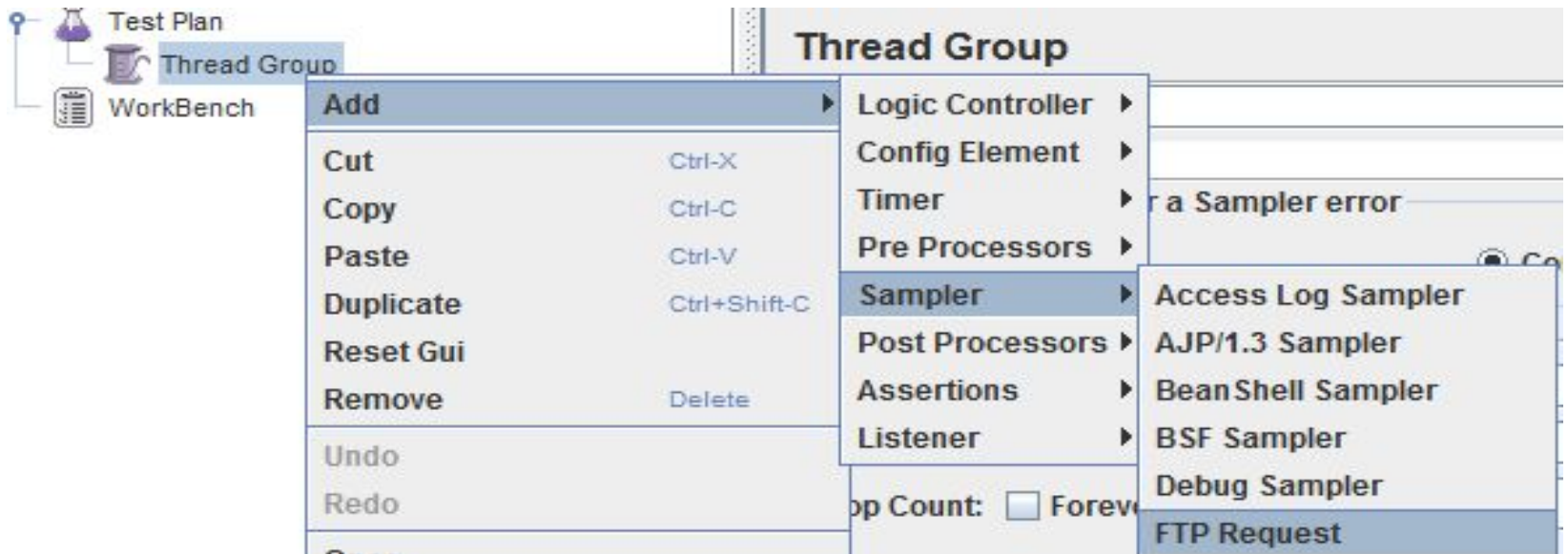
- **Access Log Sampler:** Used to generate requests based on access logs or the logs generated by the web servers
- **AJP Sampler:** Used to test the AJP service of Tomcat server
- **BeanShell Sampler:** Used to write a sampler using a BeanShell Script
- **BSF Sampler:** Used to write a sampler using a BSF Script
- **JSR223 Sampler:** Used to write a sampler using a JSR223 Script
- **Java Request Sampler:** Used to write a sampler using a Java Program

Samplers in JMeter cont..

- **JDBC Sampler:** Used to test the JDBC services
- **JMS Samplers:** Used to test the JMS services.
 - It can be used to test **point to point** as well as **publisher** and **subscriber** services under JMS
- **LDAP Requests Defaults Sampler:** Used to test the LDAP services by setting the default values for the test script
- **LDAP Extended Request Defaults Sampler:** This is same as LDAP Requests Defaults Sampler but is used for testing the extended services of LDAP
- **TCP Sampler:** Used to test the TCP services such as *telnet*, *ftp*, *https* and other services

FTP Request Sampler

- **Step1:** Right click on Test Plan and go to *Add>Threads(Users)>Thread Group*
- **Step 2:** Add a FTP Request Sampler to a test plan by right click on Thread Group and select *Add > Sampler > FTP Request*



Adding FTP Request Sampler

FTP Request Sampler cont..

- Step 3: Configure the *FTP Request* Sampler form

FTP Request

Name:

Comments:

Server Name or IP: Port Number:

Remote File:

Local File:

Local File Contents:

☒ get(RETR) ☐ put(STOR) ☐ Use Binary mode ? ☐ Save File in Response ?

Login Configuration

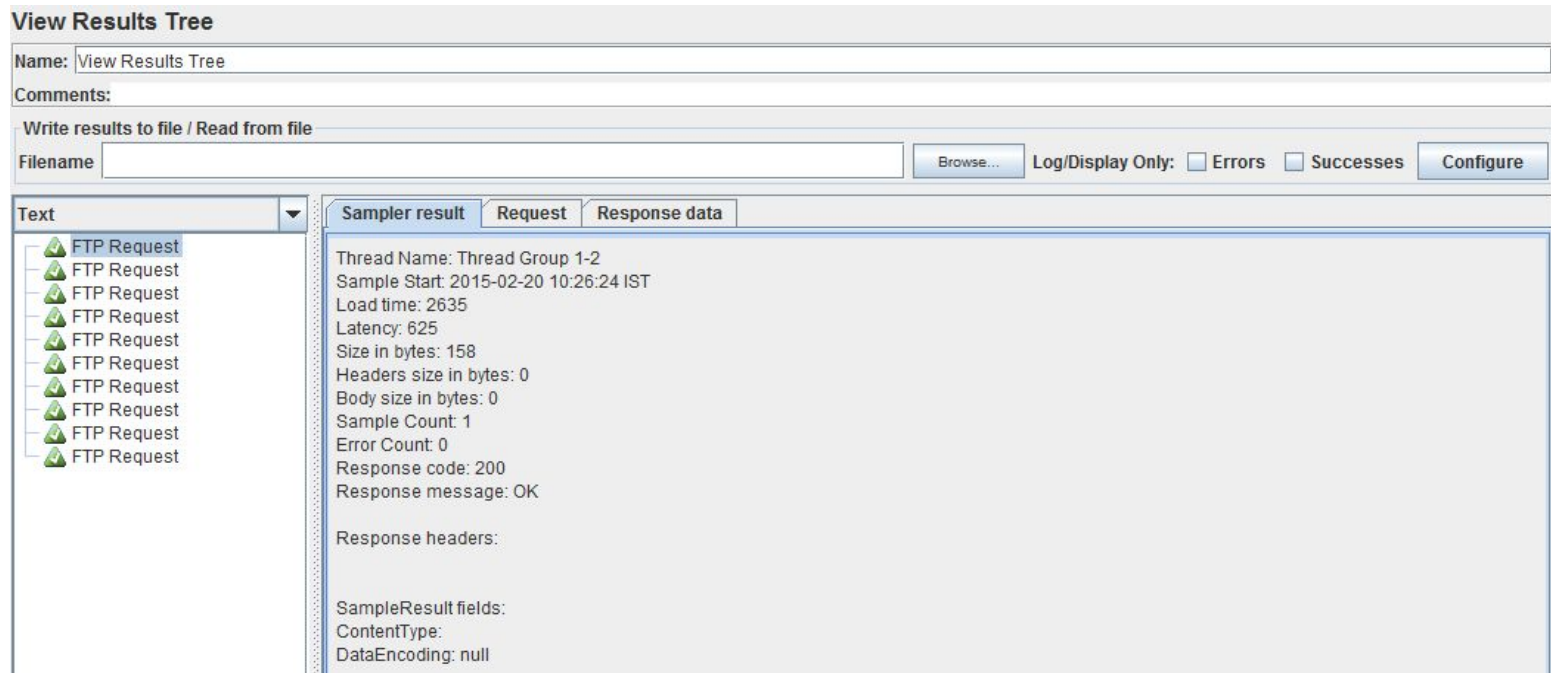
Username:

Password:

Configuring FTP Request Sampler

FTP Request Sampler cont..

- **Step 4:** Run the script and view the results in View Results Tree



View Results Tree Listener

Measuring Load

- For most of the users it becomes important to test the load on a server in terms of hits per second
- To add delays in between the work of threads JMeter provides a set of *Timers*

Timers in JMeter

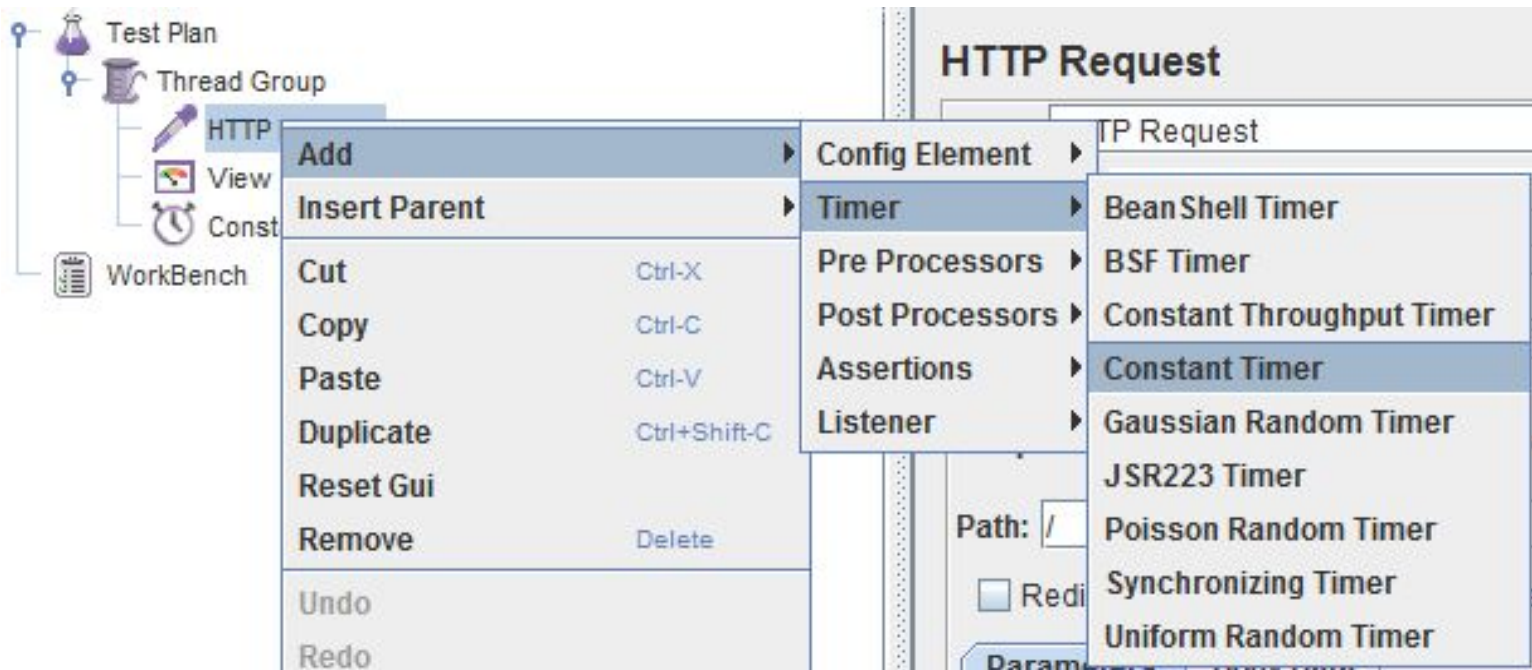
- **Constant Timer:** Used to add a constant delay in between actions of threads
- **Constant Throughput Timer:** Simulate the load in terms of throughput
- **Uniform Random Timer:** This timer introduces time gaps between the actions of each thread with random time limited by upper and lower limits of randomness.
 - There is a fixed delay and a random delay whose value suffers randomness and lastly total delay is the sum of these two
- **Gaussian Random Timer:** This timer is same as the Uniform Random Timer, but the randomness deviates based on the distribution of Gaussian curve

Timers in JMeter cont..

- **BeanShell Timer:** Used to add delay between each user request sent using BeanShell scripting
- **BSF Timer:** Used to add delay between each user request sent using BSF scripting language
- **JSR223 Timer:** Used to add delay between each user request sent using JSR223 scripting language
- **Poisson Random Timer:** Used to add delay between each threads action based on the Poisson Distribution Curve
- **Synchronizing Timer:** Used to add delay between groups of threads. It allows to add delay by synchronizing among certain number of threads

Using Constant Timer

- **Step 1:** Add a constant timer to a Thread Group by right click on Thread Group and select *Add > Timer > Constant Timer*



Adding Constant Timer

Using Constant Timer cont..

- **Step 2:** Configure the constant timer time by entering the time in *Thread Delay* field



The image shows a configuration window titled "Constant Timer". It contains three input fields: "Name:" with the value "Constant Timer", "Comments:" which is empty, and "Thread Delay (in milliseconds):" with the value "3000".

Constant Timer	
Name:	Constant Timer
Comments:	
Thread Delay (in milliseconds):	3000

Configuring Constant Timer

Using Constant Timer cont..

- **Step 3:** Now run the test script with required number of threads and view the results in *View Results in Table* listener

View Results in Table

Name:

View Results in Table

Comments:










Write results to file / Read from file

Filename

Browse...

Log/Display Only: ☐ Errors ☐ Successes

Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency
1	12:46:51.417	Thread Group 1-1	HTTP Request	1464		20237	761
2	12:46:52.017	Thread Group 1-2	HTTP Request	1507		20244	717
3	12:46:53.816	Thread Group 1-5	HTTP Request	1375		20293	840
4	12:46:53.217	Thread Group 1-4	HTTP Request	2189		20259	1103
5	12:46:52.617	Thread Group 1-3	HTTP Request	2918		20244	759
6	12:46:55.877	Thread Group 1-1	HTTP Request	2372		20204	475
7	12:46:56.523	Thread Group 1-2	HTTP Request	2648		20228	804
8	12:46:58.187	Thread Group 1-5	HTTP Request	1196		20245	467
9	12:46:58.407	Thread Group 1-4	HTTP Request	1373		20245	1076
10	12:46:58.532	Thread Group 1-3	HTTP Request	2104		20241	1065

View Results in Table Listener

Logic Controllers in JMeter

Logic controllers are used to control the test scripts. Well ordering can be done among the samplers using Logic Controllers

- **Random Order Controller:** It makes the samplers being called in random manner
- **Interleave Controller:** It makes the samplers to run in loops based on number of threads. It does not depend upon the Loop Count parameter
- **If Controller:** It allows to hit the defined sampler inside whenever the condition becomes true
- **Loop Controller:** It makes the sampler to run as many times as specified or even forever
- **Once Only Controller :** This controller makes the test script to run the components of the controller only once per thread

Logic Controllers in JMeter cont..

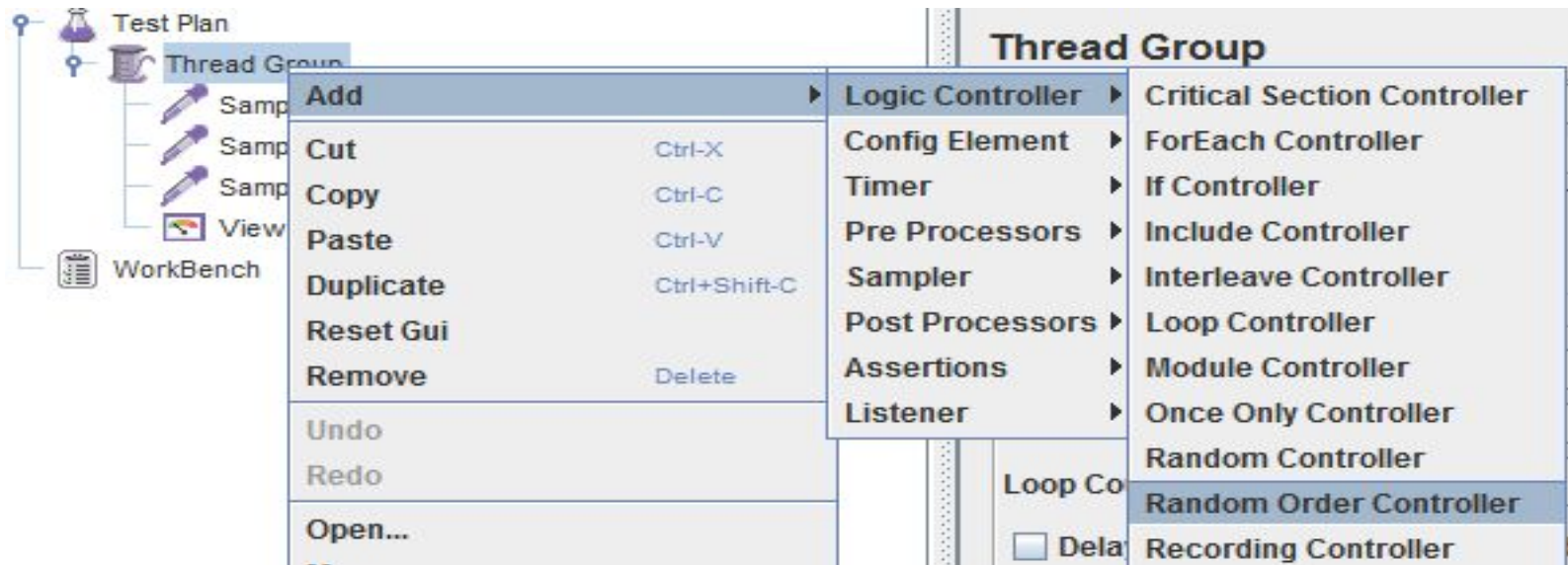
- **Random Controller** : This controller makes the choice of samplers or test elements under the controller randomly
- **Recording Controller** : Used to add the recording elements and it automatically adds the recorded elements from the JMeter Proxy Service
- **Runtime Controller** : It allows to run the test elements under it and runs for a fixed period of time
- **Switch Controller** : Used to execute the test elements under it in the same manner as Interleave Controller but it is based on the switch value
- **While Controller** : Used to loop through its test elements based on a condition

Logic Controllers in JMeter cont..

- **Critical Section Controller** : Used to serialize the execution of test elements in a test script. This is mainly used when we want certain test element to be executed at a time
- **ForEach Controller** : This is based on number of values of a variable and continuation of loop is based on value of a variable.
 - A sampler added with *ForEach Controller* executes till the ForEach variable has a value
- **Include Controller** : Used to add other test scripts in the current test script.
 - allows to add JMeter jmx files to test script
- **Module Controller** : Used to add parts of the current test plan to the current test script

Using Random Controller

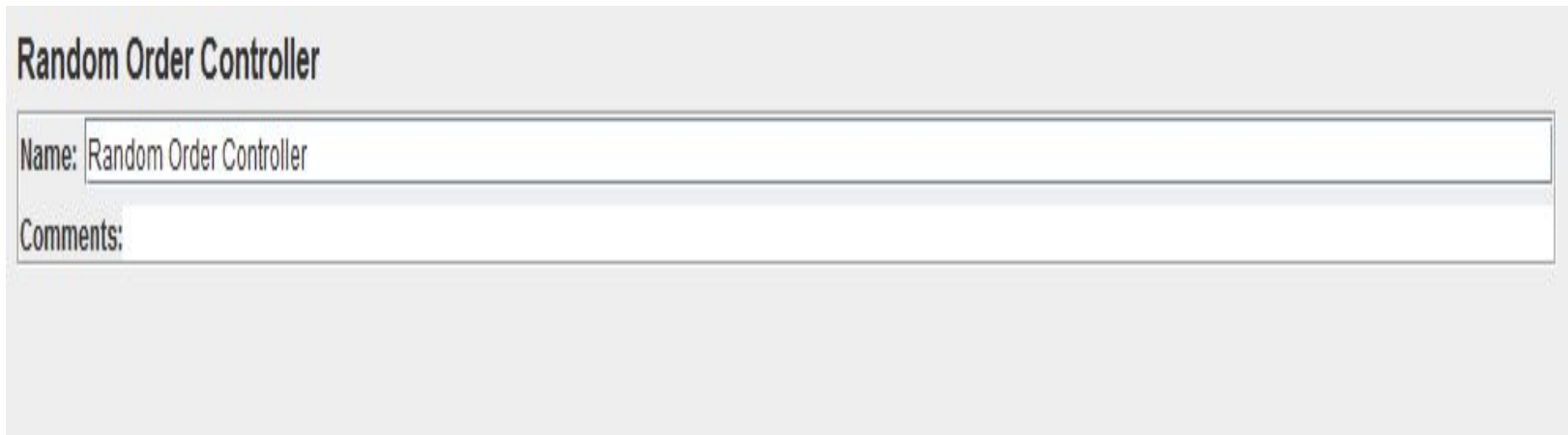
- **Step 1:** Add a random order controller to a thread group by right click on Thread Group and select *Add > Logic Controller > Random Order Controller*



Adding Random Order Controller

Using Random Controller cont..

- Step 2: The *Random Order Controller* form is configured



The screenshot shows a web form titled "Random Order Controller". It has a light gray header bar with the title. Below the header, there are two input fields. The first field is labeled "Name:" and contains the text "Random Order Controller". The second field is labeled "Comments:" and is currently empty. The form has a simple, clean design with a white background and a light gray border.

Random Order Controller Form

Using Random Controller cont..

- **Step 3:** Run the test script with required number of threads set and view the results using *View Results in Table* listener

View Results in Table							
Name: View Results in Table							
Comments:							
Write results to file / Read from file							
Filename				Browse...	Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes		Configure
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency
1	10:55:37.311	Thread Group 1-1	Sampler 1	2135		22506	1847
2	10:55:39.466	Thread Group 1-1	Sampler 3	1991		11386	1991
3	10:55:41.477	Thread Group 1-1	Sampler 2	6322		62375	1691
4	10:55:47.803	Thread Group 1-1	Sampler 2	2306		62346	445
5	10:55:50.112	Thread Group 1-1	Sampler 1	566		22457	427
6	10:55:50.681	Thread Group 1-1	Sampler 3	1356		11386	1105
7	10:55:52.041	Thread Group 1-1	Sampler 2	3407		62417	826
8	10:55:55.452	Thread Group 1-1	Sampler 1	865		22457	468
9	10:55:56.320	Thread Group 1-1	Sampler 3	2		1939	0
10	10:55:56.412	Thread Group 1-1	Sampler 2	3447		62454	986
11	10:55:59.863	Thread Group 1-1	Sampler 1	1634		22450	366
12	10:56:01.500	Thread Group 1-1	Sampler 3	1837		11386	1698
13	10:56:03.341	Thread Group 1-1	Sampler 1	476		22449	257
14	10:56:03.818	Thread Group 1-1	Sampler 3	1839		11385	591
15	10:56:05.660	Thread Group 1-1	Sampler 2	5018		62336	458

View Results in Table Listener

Error Handling

- JMeter allows to create dynamic test scripts based on checking the responses from the server
- Through assertions, the response from the server can be checked in terms of size, duration, finding patterns etc.
- Other functionalities of JMeter assertions include testing the response by checking whether the HTML/XML document received from server contains any warnings or errors

Assertions in JMeter

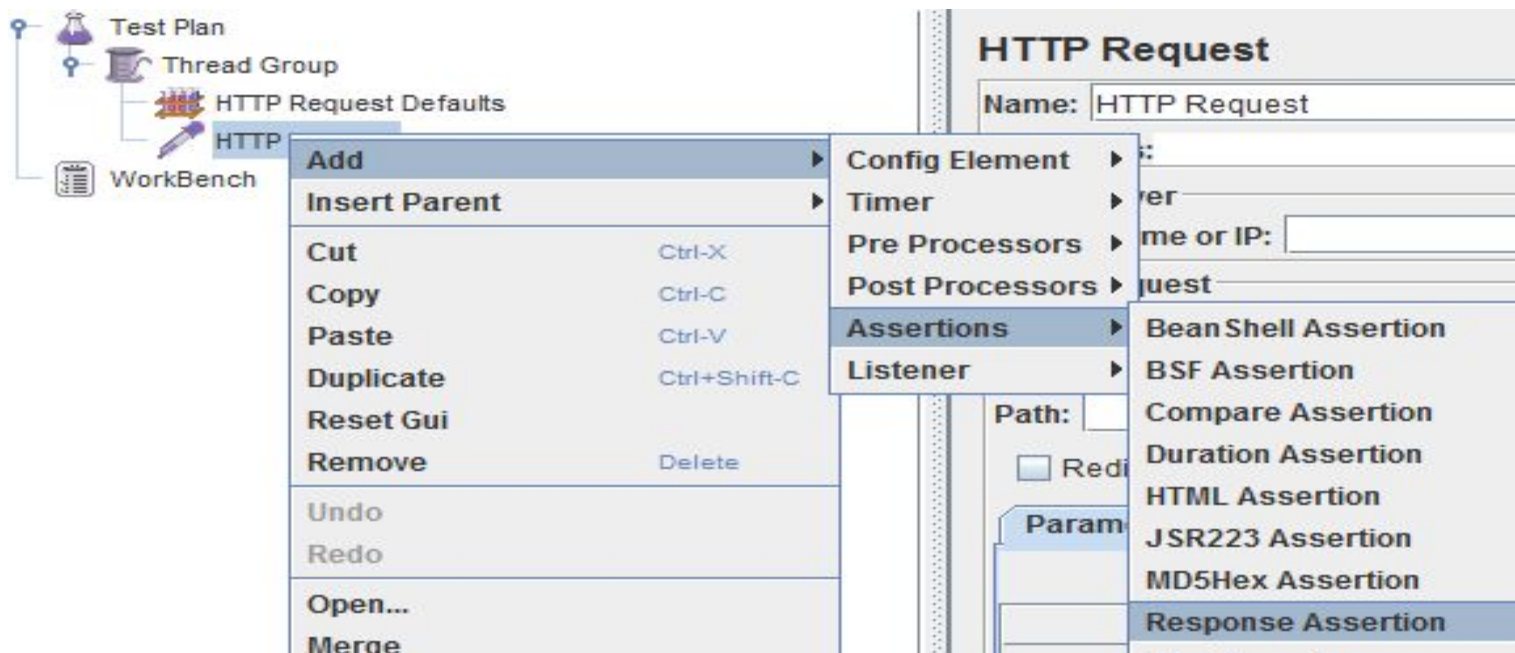
- **Response Assertion:** Used to find any pattern or response code from the response of the server
 - Generally, Google website has some response codes which include 404,200,302. The response code 404 means error, 200 represents everything is fine and 302 represents redirection to another URL
- **Duration Assertion:** Used to compare the response time with the specified time configured in *Response Assertion* form
- **HTML Assertion:** Used to check the syntax of the received HTML document and can also alert whenever there is any HTML warnings
- **Size Assertion:** Used to check the size of the response in terms of bytes

Assertions in JMeter cont..

- **BeanShell Assertion:** Used to check the response by writing a BeanShell script
- **BSF Assertion:** Used to check the response by writing a BSF script
- **JSR223 Assertion:** Used to check the response by writing a JSR223 script
- **Compare Assertion:** Used to compare the response within its scope, it can include time and comparisons filter. It consumes much CPU overhead
- **MD5Hex Assertion:** Used to check the MD5 hash of the response data
- **XML Schema Assertion:** Used to validate the response against a given XML schema

Adding Response Assertion

- **Step 1:** Add a Response Assertion to a sampler by right click on any Sampler and select *Add > Assertions > Response Assertion*



Adding Response Assertion

Adding Response Assertion cont..

- Step 2: Configure the *Response Assertion* form

The screenshot shows the 'Response Assertion' configuration form in JMeter. The form is divided into several sections:

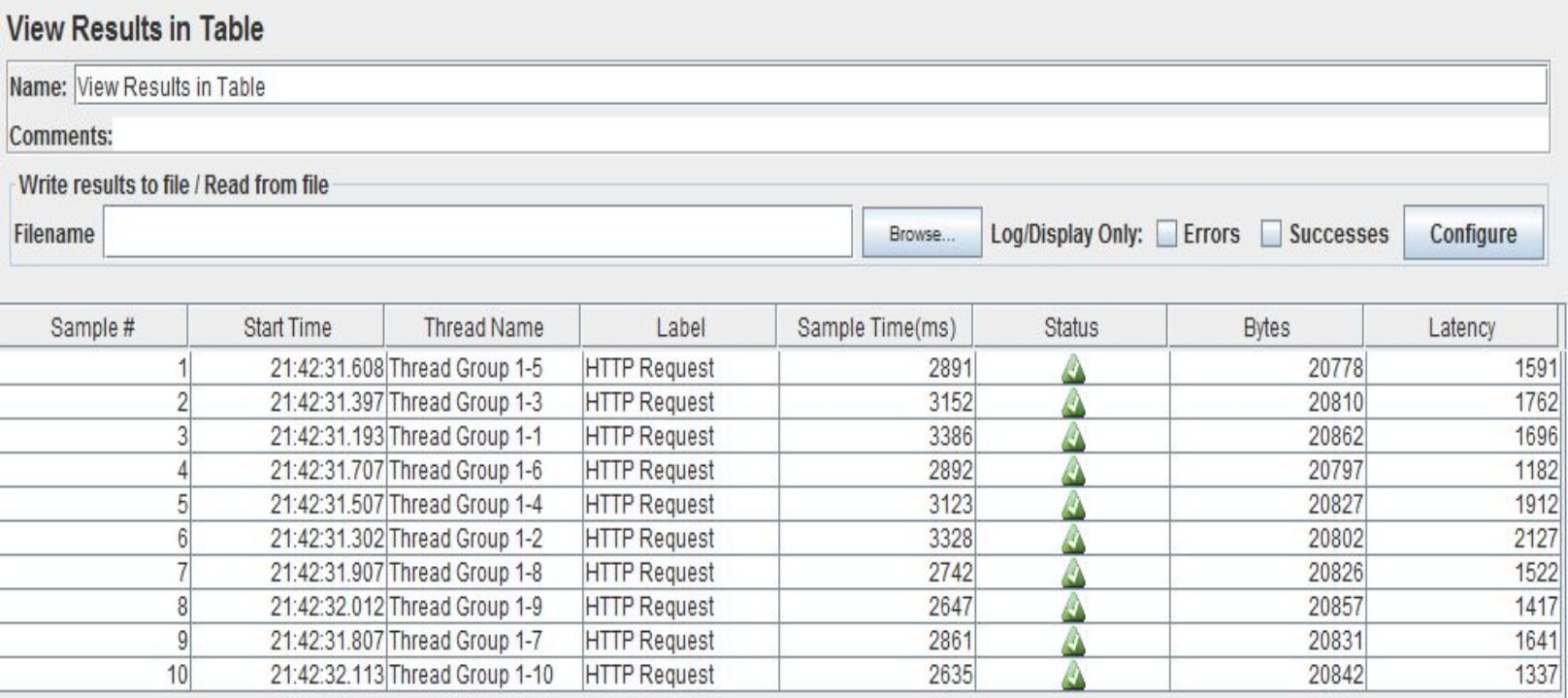
- Response Assertion**: The title of the form.
- Name:** A text field containing 'Response Assertion'.
- Comments:** An empty text area.
- Apply to:** A section with radio buttons for 'Main sample and sub-samples', 'Main sample only' (selected), 'Sub-samples only', and 'JMeter Variable'. There is an empty text field next to 'JMeter Variable'.
- Response Field to Test**: A section with radio buttons for 'Text Response', 'Document (text)', 'URL Sampled', 'Response Code' (selected), 'Response Message', 'Response Headers', and 'Ignore Status'.
- Pattern Matching Rules**: A section with radio buttons for 'Contains', 'Matches', 'Equals', 'Substring' (selected), and 'Not'.
- Patterns to Test**: A table with one row containing the value '200'.

Patterns to Test
200

Response Assertion Form

Adding Response Assertion cont..

- Step 3: Run the script and view the results in *View Results in Table* Listener

- 

The screenshot shows the 'View Results in Table' listener configuration window. It includes fields for Name, Comments, and a section for writing results to a file. Below the configuration is a table displaying 10 test results, each with a sample number, start time, thread name, label, sample time, status (indicated by a green triangle icon), bytes, and latency.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency
1	21:42:31.608	Thread Group 1-5	HTTP Request	2891		20778	1591
2	21:42:31.397	Thread Group 1-3	HTTP Request	3152		20810	1762
3	21:42:31.193	Thread Group 1-1	HTTP Request	3386		20862	1696
4	21:42:31.707	Thread Group 1-6	HTTP Request	2892		20797	1182
5	21:42:31.507	Thread Group 1-4	HTTP Request	3123		20827	1912
6	21:42:31.302	Thread Group 1-2	HTTP Request	3328		20802	2127
7	21:42:31.907	Thread Group 1-8	HTTP Request	2742		20826	1522
8	21:42:32.012	Thread Group 1-9	HTTP Request	2647		20857	1417
9	21:42:31.807	Thread Group 1-7	HTTP Request	2861		20831	1641
10	21:42:32.113	Thread Group 1-10	HTTP Request	2635		20842	1337

View Results in Table Listener

Conclusion

- JMeter is an open source tool
- Used to perform load testing in a web server or group of web servers
- Provides a number of features such as Timers, Controllers, Recording Scripts, Assertions, and plug-ins etc. for load testing in web servers
- Used to perform load testing of variety of web services using its features

Thank You

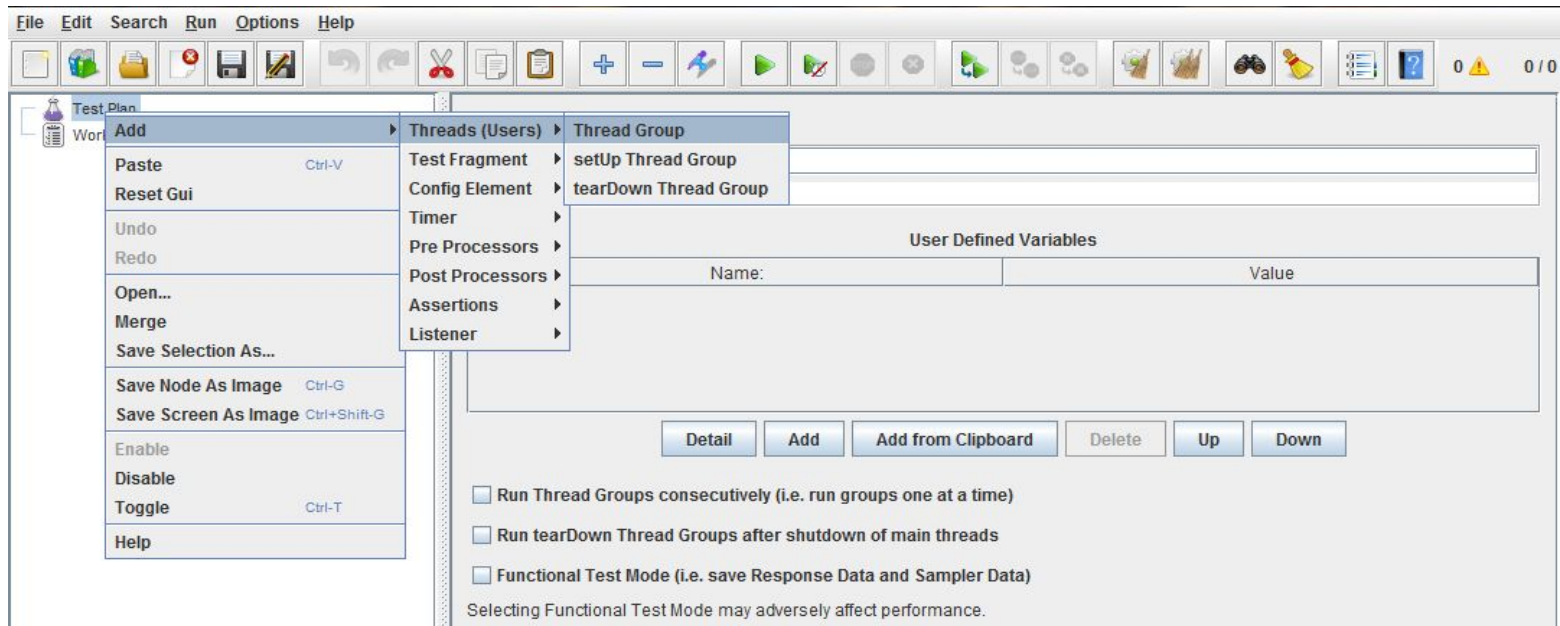


Exercise 1

Create a JMeter Test Script to test the load on *www.facebook.com* with 100 users and 3 loops each.

Solution 1

- Open JMeter and right click on Test Plan and select *Add > Threads (Users) > Thread Group*



Create Thread Group

Solution 1 cont..

- Set Number of Threads(users) to 100 and Loop Count to 3

Thread Group

Name:

Comments:

Action to be taken after a Sampler error

☒ Continue ☐ Start Next Thread Loop ☐ Stop Thread ☐ Stop Test ☐ Stop Test Now

Thread Properties

Number of Threads (users):

Ramp-Up Period (in seconds):

Loop Count: ☐ Forever

☐ Delay Thread creation until needed

☐ Scheduler

Thread Group Form

Solution 1 cont..

- Add a HTTP Request sampler, set the server name to *www.facebook.com* and path to /

The screenshot shows the 'HTTP Request' configuration form in JMeter. The 'Name' field is set to 'HTTP Request'. The 'Web Server' section has 'Server Name or IP' set to 'www.facebook.com'. The 'Timeouts (milliseconds)' section has 'Connect' and 'Response' fields. The 'HTTP Request' section has 'Implementation' as a dropdown, 'Protocol [http]' as a text field, 'Method' as a dropdown set to 'GET', and 'Content encoding' as a text field. The 'Path' field is set to '/'. At the bottom, there are checkboxes for 'Redirect Automatically', 'Follow Redirects' (checked), 'Use KeepAlive' (checked), 'Use multipart/form-data for POST', and 'Browser-compatible headers'.

HTTP Request	
Name:	HTTP Request
Comments:	
Web Server	
Server Name or IP:	www.facebook.com
Port Number:	
Timeouts (milliseconds)	
Connect:	
Response:	
HTTP Request	
Implementation:	
Protocol [http]:	
Method:	GET
Content encoding:	
Path: /	
<input type="checkbox"/> Redirect Automatically <input checked="" type="checkbox"/> Follow Redirects <input checked="" type="checkbox"/> Use KeepAlive <input type="checkbox"/> Use multipart/form-data for POST <input type="checkbox"/> Browser-compatible headers	

HTTP Request Sampler Form

Solution 1 cont..

- Add Listener to collect the results. Right click on Thread Group, select *Add>Listener>View Results in Table*
- Save the test script using **ctrl + S** or **File □ Save**
- Run the test script using **ctrl + R** or from File menu

Solution 1 cont..

- View the results from **View Results in Table** under the test script tree in Test Plan Pane

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Browse...

Log/Display Only: ☐ Errors ☐ Success

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes
1	10:53:59.167	Thread Group 1-...	HTTP Request	323	Success	
2	10:53:59.218	Thread Group 1-...	HTTP Request	272	Success	
3	10:53:59.085	Thread Group 1-...	HTTP Request	405	Success	
4	10:53:59.265	Thread Group 1-...	HTTP Request	226	Success	
5	10:53:59.212	Thread Group 1-...	HTTP Request	278	Success	
6	10:53:59.247	Thread Group 1-...	HTTP Request	243	Success	
7	10:53:59.122	Thread Group 1-...	HTTP Request	370	Success	
8	10:53:59.014	Thread Group 1-...	HTTP Request	476	Success	
9	10:53:59.168	Thread Group 1-...	HTTP Request	321	Success	
10	10:53:59.048	Thread Group 1-1	HTTP Request	441	Success	
11	10:53:59.093	Thread Group 1-...	HTTP Request	396	Success	
12	10:53:59.123	Thread Group 1-5	HTTP Request	369	Success	
13	10:53:59.047	Thread Group 1-...	HTTP Request	442	Success	
14	10:53:59.244	Thread Group 1-...	HTTP Request	245	Success	
15	10:53:59.257	Thread Group 1-...	HTTP Request	237	Success	
16	10:53:59.030	Thread Group 1-...	HTTP Request	459	Success	
17	10:53:58.999	Thread Group 1-...	HTTP Request	490	Success	
18	10:53:58.998	Thread Group 1-...	HTTP Request	491	Success	
19	10:53:59.261	Thread Group 1-...	HTTP Request	228	Success	
20	10:53:59.051	Thread Group 1-...	HTTP Request	438	Success	
21	10:53:59.151	Thread Group 1-...	HTTP Request	347	Success	

View Results in Table Listener

Exercise 2

Create a script to record visiting a online shopping or social network site and play it with **100 threads** and **3 loops** per thread and view the results using View Results Tree.

Solution 2 cont..

- Go to File menu and select **Templates**. In the Templates dialog box the recording script is selected by default and click on **Create button**



Templates Dialog Box

Solution 2 cont..

- Now click on **HTTP(S) Test Script Recorder** in **WorkBench**, set the proxy port and start the JMeter proxy service by clicking on the **Start** button

The screenshot shows the 'HTTP(S) Test Script Recorder' configuration window. It includes fields for Name, Comments, Global Settings (Port, HTTPS Domains), Test plan content (Target Controller, Grouping), HTTP Sampler settings (Type, Redirect Automatically, Follow Redirects, Use KeepAlive, Retrieve All Embedded Resources), Content-type filter (Include, Exclude), URL Patterns to Include, URL Patterns to Exclude, and a checkbox for 'Notify Child Listeners of filtered samplers'. At the bottom are 'Start', 'Stop', and 'Restart' buttons.

HTTP(S) Test Script Recorder

Name: HTTP(S) Test Script Recorder

Comments:

Global Settings

Port: 8888 HTTPS Domains :

Test plan content

Target Controller: Use Recording Controller

Grouping: Put each group in a new transaction controller ☒ Capture HTTP Headers ☐ Add Assertions ☒ Regex matching

HTTP Sampler settings

Type: ☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Retrieve All Embedded Resources

Content-type filter

Include: Exclude:

URL Patterns to Include

URL Patterns to Include

Add Delete Add from Clipboard

URL Patterns to Exclude

URL Patterns to Exclude

(?i).*\.(bmp|css|js|gif|ico|jpe?g|png|swf|woff){1?}.*
(?i).*\.(bmp|css|js|gif|ico|jpe?g|png|swf|woff)

Add Delete Add from Clipboard Add suggested Excludes

Notify Child Listeners of filtered samplers

☒ Notify Child Listeners of filtered samplers

Start Stop Restart

HTTP Test Script Recorder Form

Solution 2 cont..

- Change the proxy setting of your browser to point to the JMeter proxy server. Set the proxy server address to the address of the JMeter system and in this case the JMeter address is the localhost and port is 8888
- Go to *Options > Advanced Tab > Network >* and configure the proxy settings and *Use this proxy server for all protocols Settings*



The screenshot shows the 'Manual proxy configuration' section of a browser's network settings. It includes fields for HTTP, SSL, FTP, and SOCKS proxies, all set to 127.0.0.1 on port 8888. The checkbox 'Use this proxy server for all protocols' is checked. The 'No Proxy for' field contains 'localhost, 127.0.0.1'. The SOCKS version is set to v5.

☒ **Manual proxy configuration:**

HTTP Proxy: 127.0.0.1 Port: 8888

☒ Use this proxy server for all protocols

SSL Proxy: 127.0.0.1 Port: 8888

FTP Proxy: 127.0.0.1 Port: 8888

SOCKS Host: 127.0.0.1 Port: 8888

☐ SOCKS v4 ☒ SOCKS v5

No Proxy for: localhost, 127.0.0.1

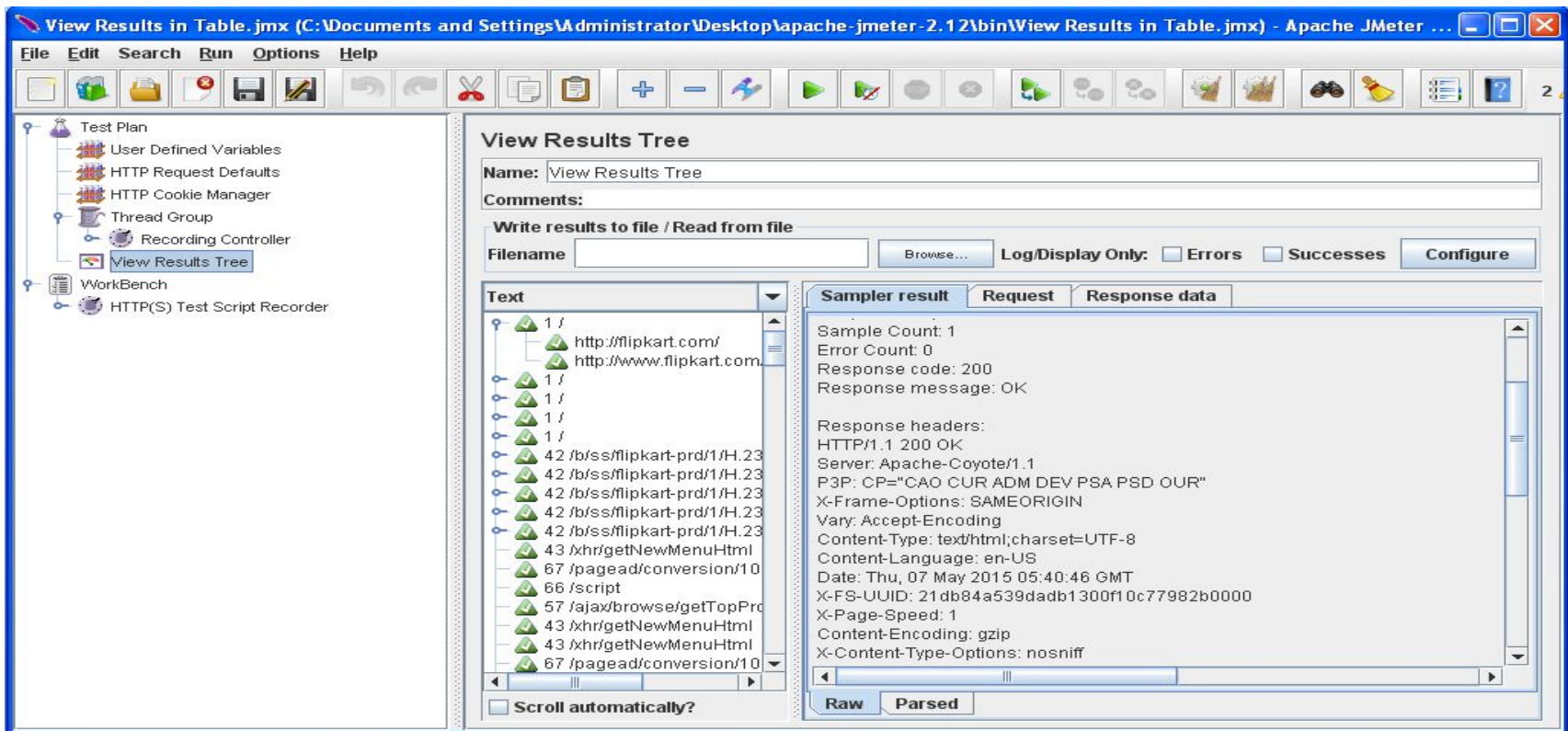
Example: mozilla.org, .net.nz, 192.168.1.0/24

Solution 2 cont..

- Now we would be recording login to a site and logout from it and use any online site e.g. www.flipkart.com for testing. Visit to login and then logout
- Before playing the script, modify the number of threads as per requirements

Solution 2 cont..

- Play the script and the results can be seen from the **View Results Tree** listener from the Test Pane



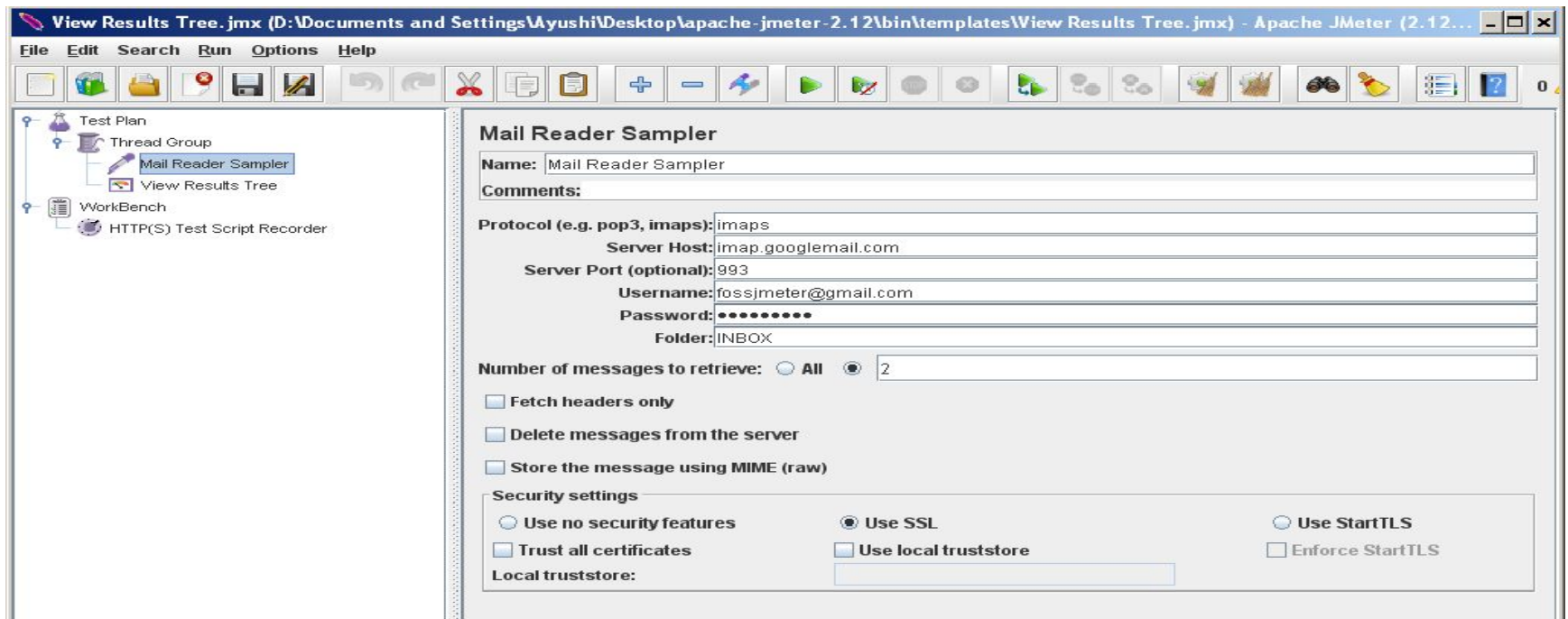
View Results in Tree Listener

Exercise 3

Create a test script to read 30 mails from the Inbox of your Gmail account.

Solution 3 cont..

- Create a Mail Reader Sampler and set the field **Number of Messages to Retrieve** to 2
- Run the script with 15 threads so that each thread downloads 2 mails and totals to 30 mails



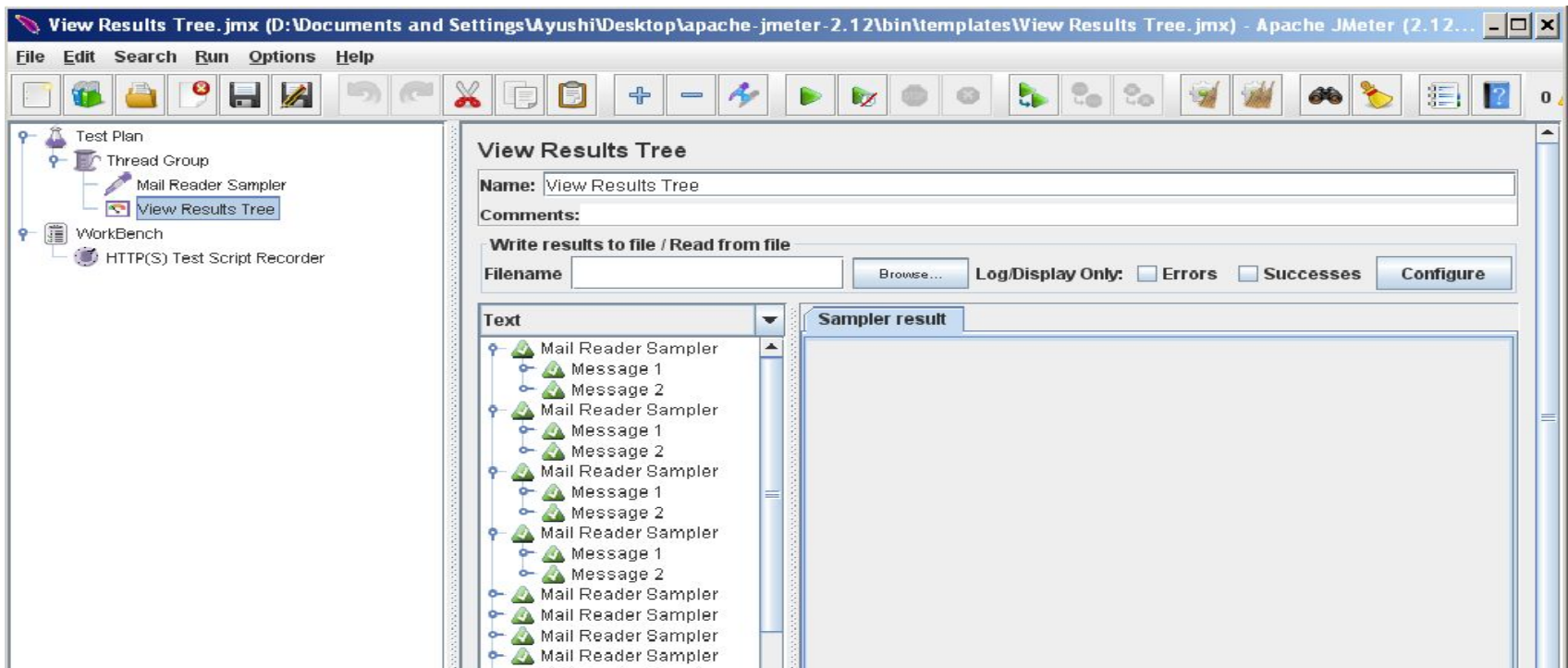
The screenshot shows the Apache JMeter interface with the 'Mail Reader Sampler' configuration form open. The form is titled 'Mail Reader Sampler' and contains the following fields and options:

- Name:** Mail Reader Sampler
- Comments:**
- Protocol (e.g. pop3, imaps):** imaps
- Server Host:** imap.googlemail.com
- Server Port (optional):** 993
- Username:** fossjmeter@gmail.com
- Password:** ••••••••
- Folder:** INBOX
- Number of messages to retrieve:** ☐ All ☒ 2
- ☐ Fetch headers only
- ☐ Delete messages from the server
- ☐ Store the message using MIME (raw)
- Security settings:**
 - ☐ Use no security features
 - ☒ Use SSL
 - ☐ Use StartTLS
 - ☐ Trust all certificates
 - ☐ Use local truststore
 - ☐ Enforce StartTLS
- Local truststore:**

Mail Reader Sampler Form

Solution 3 cont..

- Play the script and the results can be seen from the **View Results Tree** element from the Test Pane



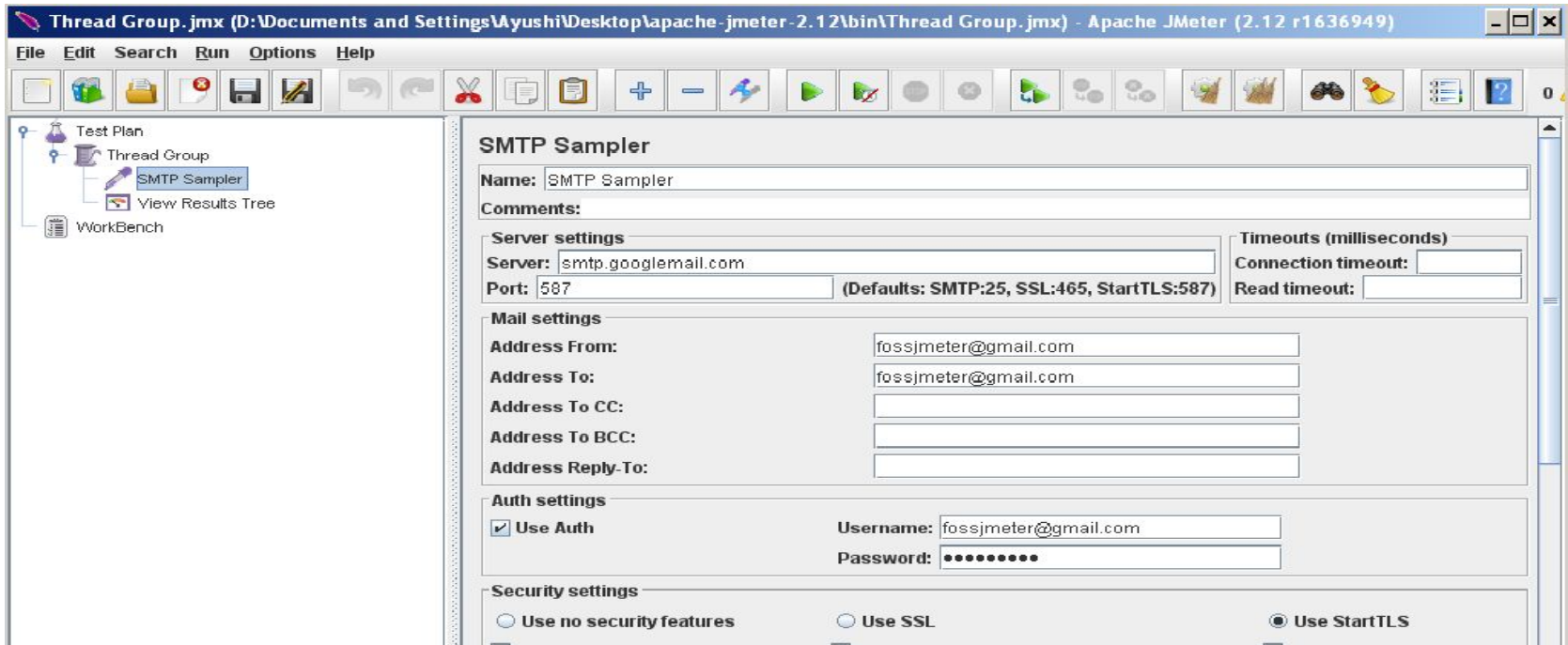
View Results Tree Listener

Exercise 4

Create a test script to send mail to a Gmail user using the same Gmail account

Solution 4

- Add a SMTP Sampler to a Thread Group
- Now configure the SMTP Sampler



The screenshot shows the Apache JMeter interface with the SMTP Sampler configuration window open. The window title is "Thread Group.jmx (D:\Documents and Settings\Ayushi\Desktop\apache-jmeter-2.12\bin\Thread Group.jmx) - Apache JMeter (2.12 r1636949)". The left sidebar shows a tree view with "Test Plan", "Thread Group", "SMTP Sampler", "View Results Tree", and "WorkBench". The main panel displays the configuration for the "SMTP Sampler".

SMTP Sampler

Name: SMTP Sampler

Comments:

Server settings

Server: smtp.googlemail.com

Port: 587 (Defaults: SMTP:25, SSL:465, StartTLS:587)

Timeouts (milliseconds)

Connection timeout:

Read timeout:

Mail settings

Address From: fossjmeter@gmail.com

Address To: fossjmeter@gmail.com

Address To CC:

Address To BCC:

Address Reply-To:

Auth settings

☒ Use Auth

Username: fossjmeter@gmail.com

Password:

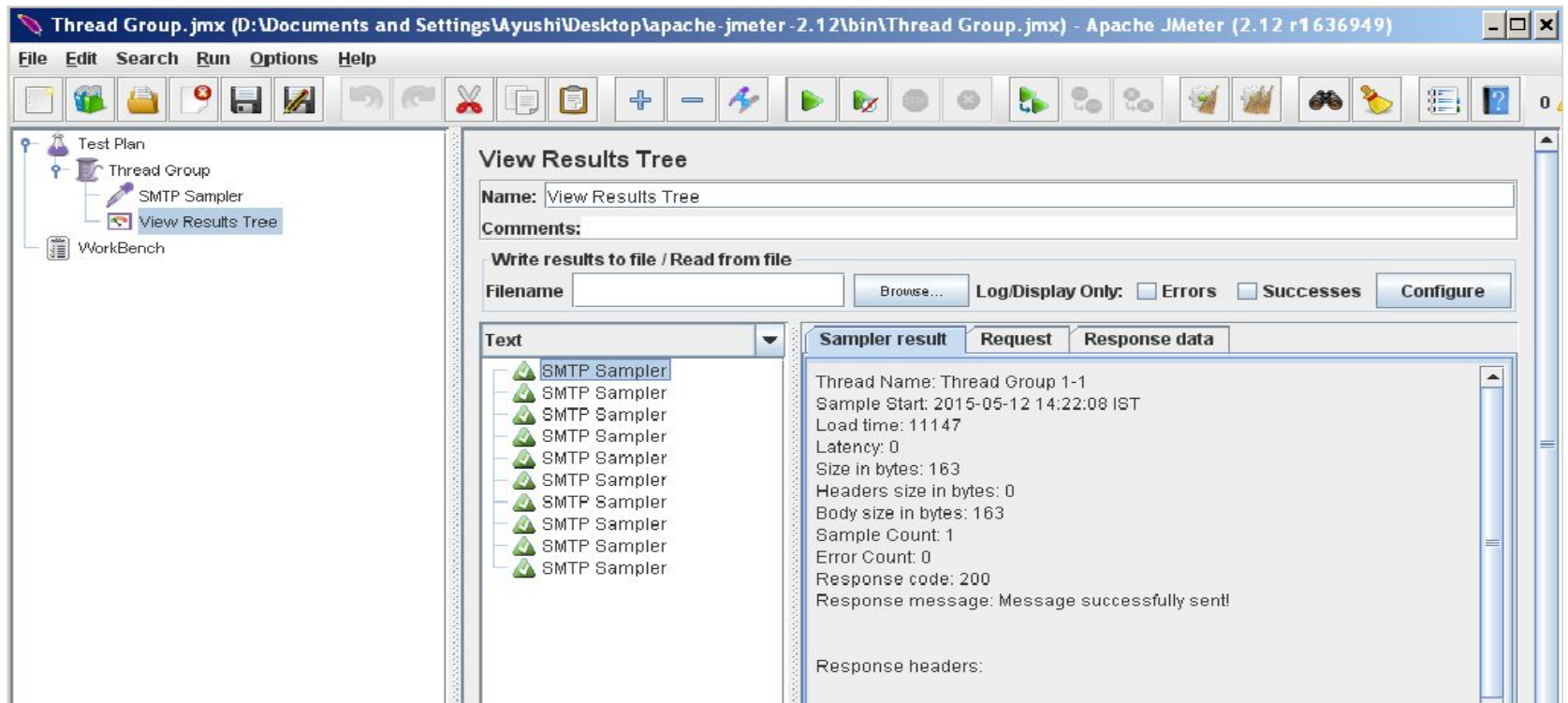
Security settings

☐ Use no security features ☐ Use SSL ☒ Use StartTLS

SMTP Sampler Form

Solution 4 cont...

- Run the script and view the results in View Results Tree



View Results Tree Listener

Exercise 5

Add a **While Controller** to test script which tests the HTTP service of Google web site which is based on the condition that it will execute the next thread when the previous thread execution is successful.

Solution 5 cont..

- Create a basic test script to hit HTTP service of Google at www.google.co.in
- Now add a **While Controller** for the HTTP Sampler and configure the condition of the While Controller to *`${JmeterThread:last_sample_ok}`*

While Controller

Name:	While Controller
Comments:	
Condition (function or variable)	<code>\${JmeterThread:last_sample_ok}</code>

While Controller Form

Solution 5 cont..

- Run the test script with required number of threads

The screenshot shows the Apache JMeter interface with the 'View Results in Table' listener selected. The test plan on the left includes a Thread Group with an HTTP Request and a While Controller. The table displays 10 samples, all successful, with sample times ranging from 415 to 539 ms. The average sample time is 438 ms.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes
1	17:13:17.216	Thread Group 1-1	HTTP Request	438	✓	
2	17:13:17.323	Thread Group 1-2	HTTP Request	415	✓	
3	17:13:17.435	Thread Group 1-3	HTTP Request	434	✓	
4	17:13:17.519	Thread Group 1-4	HTTP Request	416	✓	
5	17:13:17.633	Thread Group 1-5	HTTP Request	430	✓	
6	17:13:17.714	Thread Group 1-6	HTTP Request	429	✓	
7	17:13:17.837	Thread Group 1-7	HTTP Request	417	✓	
8	17:13:17.937	Thread Group 1-8	HTTP Request	440	✓	
9	17:13:18.133	Thread Group 1-10	HTTP Request	424	✓	
10	17:13:18.035	Thread Group 1-9	HTTP Request	539	✓	

Scroll automatically? ☐ Child samples? ☐ No of Samples 10 Latest Sample 539 Average 438

View Results in Table Listener

Exercise 6

Create a script which will raise a warning when the response from the server gets longer by 2000 milliseconds.

Solution 6

- Create a basic test script to hit HTTP service of Google at www.google.co.in
- Now add a **Duration Assertion** for the http sampler and configure the **Duration in milliseconds** of Assertion form to **2000**



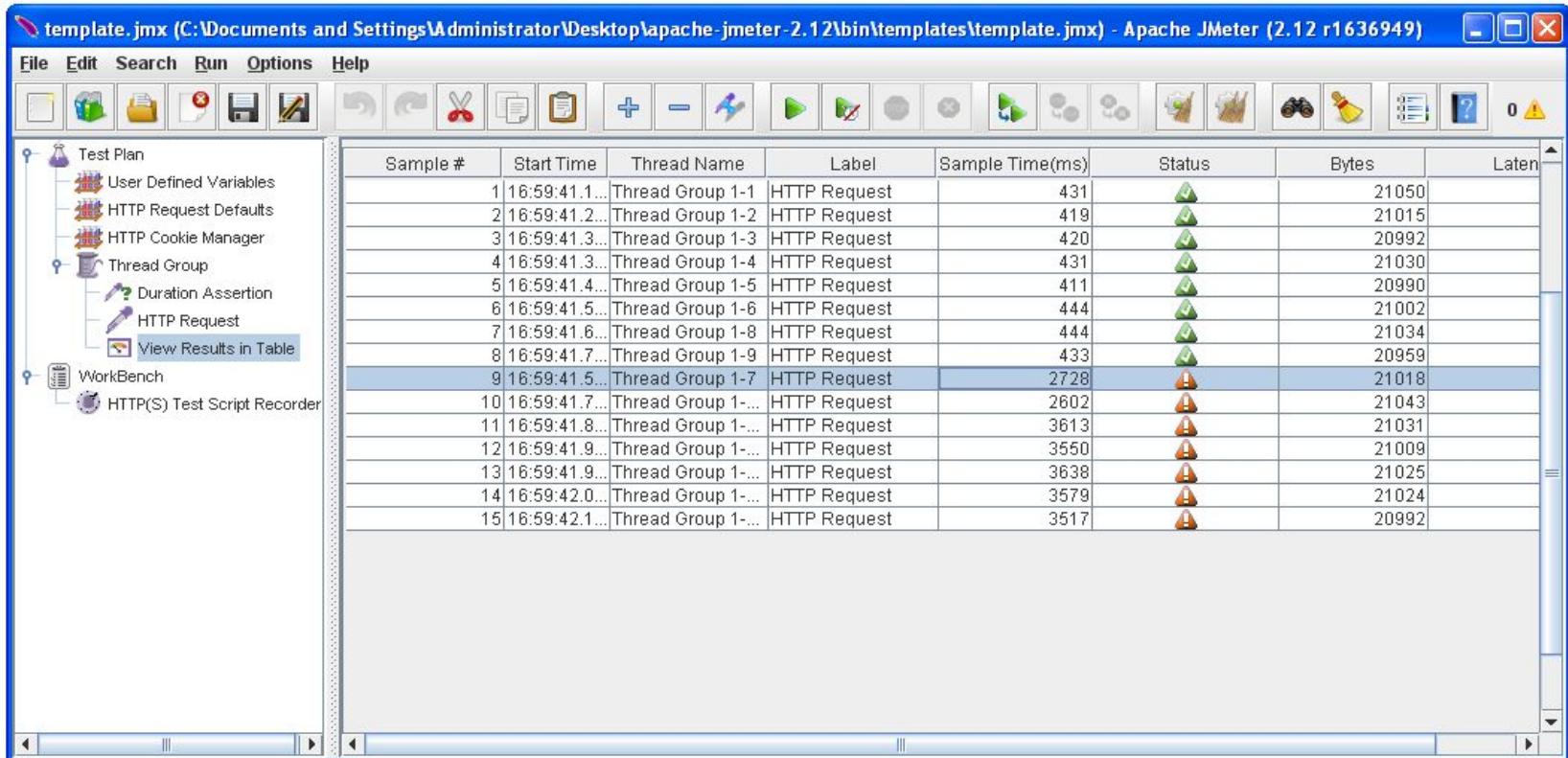
The image shows a screenshot of the 'Duration Assertion' configuration form in JMeter. The form has a title bar 'Duration Assertion'. Below it, there is a 'Name' field containing 'Duration Assertion' and a 'Comments' field. The 'Apply to' section contains three radio buttons: 'Main sample and sub-samples' (unselected), 'Main sample only' (selected), and 'Sub-samples only' (unselected). The 'Duration to Assert' section contains a 'Duration in milliseconds' field with the value '2000'.

Duration Assertion	
Name:	Duration Assertion
Comments:	
Apply to:	
<input type="radio"/> Main sample and sub-samples	<input checked="" type="radio"/> Main sample only
<input type="radio"/> Sub-samples only	
Duration to Assert	
Duration in milliseconds:	2000

Duration Assertion Form

Solution 6 cont..

- Run the test script with required number of threads



The screenshot shows the Apache JMeter 2.12 interface. The title bar indicates the file is 'template.jmx' located at 'C:\Documents and Settings\Administrator\Desktop\apache-jmeter-2.12\bin\templates\template.jmx'. The menu bar includes File, Edit, Search, Run, Options, and Help. The toolbar contains various icons for file operations, test execution, and monitoring. The left sidebar shows a tree view of the test plan components: Test Plan, User Defined Variables, HTTP Request Defaults, HTTP Cookie Manager, Thread Group, Duration Assertion, HTTP Request, View Results in Table, WorkBench, and HTTP(S) Test Script Recorder. The main area displays a table of test results for 15 samples. The table has columns for Sample #, Start Time, Thread Name, Label, Sample Time(ms), Status, Bytes, and Latency. Samples 1 through 8 show successful HTTP requests with green status icons. Sample 9 is highlighted in blue and shows a status icon with an exclamation mark. Samples 10 through 15 also show status icons with exclamation marks, indicating some form of error or warning.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency
1	16:59:41.1...	Thread Group 1-1	HTTP Request	431	✓	21050	
2	16:59:41.2...	Thread Group 1-2	HTTP Request	419	✓	21015	
3	16:59:41.3...	Thread Group 1-3	HTTP Request	420	✓	20992	
4	16:59:41.3...	Thread Group 1-4	HTTP Request	431	✓	21030	
5	16:59:41.4...	Thread Group 1-5	HTTP Request	411	✓	20990	
6	16:59:41.5...	Thread Group 1-6	HTTP Request	444	✓	21002	
7	16:59:41.6...	Thread Group 1-8	HTTP Request	444	✓	21034	
8	16:59:41.7...	Thread Group 1-9	HTTP Request	433	✓	20959	
9	16:59:41.5...	Thread Group 1-7	HTTP Request	2728	⚠	21018	
10	16:59:41.7...	Thread Group 1-...	HTTP Request	2602	⚠	21043	
11	16:59:41.8...	Thread Group 1-...	HTTP Request	3613	⚠	21031	
12	16:59:41.8...	Thread Group 1-...	HTTP Request	3550	⚠	21009	
13	16:59:41.9...	Thread Group 1-...	HTTP Request	3638	⚠	21025	
14	16:59:42.0...	Thread Group 1-...	HTTP Request	3579	⚠	21024	
15	16:59:42.1...	Thread Group 1-...	HTTP Request	3517	⚠	20992	

View Results in Table Listener