JMeter

Performance Testing Training with JMeter

Prepared By Prashanth Kumar
Performance Architect

Agenda

1. Introduction to performance Testing

Overview of Performance Testing

Performance testing is an non-functional testing to determine the system responsiveness

Performance testing is in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload

Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage.

Speed - Determines whether the application responds quickly

Scalability - Determines maximum user load the software application can handle.

Stability - Determines if the application is stable under varying loads

1. Introduction to performance Testing

Purpose of Performance Testing

1.More importantly, performance testing uncovers what needs to be improved before the product goes to market.

2. Without performance testing, software is likely to suffer from issues such as running slow while several users use it simultaneously.

3.Inconsistencies across different operating systems and poor usability

1. Introduction to performance Testing

Key Types of Performance Testing

Load Testing: It help us to study the behavior of the application under various loads. The Main Parameter to focus is Response Time & CPU Utilization

Stress Testing: It help us to observer the stability of the application the main Intension is to identify the breaking point of the system & through put

Endurance Testing: It reveals about **memory utilization** when you load your test for prolong executions

1. Introduction to performance Testing

Goal of Performance Testing.

- Compare performance characteristics of system configurations.
- Discover which part of the application performs poorly and under what conditions.
- Finding the source of performance problems.
- Support system tuning.

1. Introduction to performance Testing

Pre-Requisites Performance

- Stable build free from all major functional defects.
- Performance testing environment similar to production environment
- No other testing should be performed while performance testing
- Conduct performance testing before going to live.
- Complete understanding and knowledge of the application
- Servers information's.
- Test data preparation, PT Requirements gathering
- Performance Acceptance criteria

1. Introduction to performance Testing

Why to use performance Testing tool?

Recommend the tool that best serves the organization's purpose to test their web applications before deployment.

Tools such as profilers to measure what parts of a device or software contribute most to the poor performance, or to establish throughput levels (and thresholds) for maintained acceptable response times

1. Introduction to performance Testing

Challenged with Performance Testing

✓ Infrastructure setup

✓ Collection and analysis of Huge data

✓ Client environment

✓ Testing on live servers

✓ Team effort is required (product vendors, architects, developers, testers, DB Admin and N/W Admin.)

✓ Identifying the Root cause

2. Get Started

Contents

2. Get Started

What is JMeter?

JMeter is an Apache Jakarta project that can be used as a load testing tool for analyzing and measuring the performance of a variety of services, with a focus on web applications

The **Apache JMeter** is pure Java **open source** software

JMeter is fully Multi – threading frame work

Plat form independent and Easy Instillation

Friendly GUI and unlimited testing capabilities.

2. Get Started

What can you test in JMeter

Web: Http, Http’s sites

Web Services: SOAP/XML-RPC

Data Base via JDBC drivers.

Directory: LDAP

Massage oriented services via JMS

Services : SMTP, POP3,FTP services.

2. Get Started

Installing JMeter

- Install java 1.6 or above
- Download JMeter
- http://jmeter.apache.org/download_jmeter.cgi

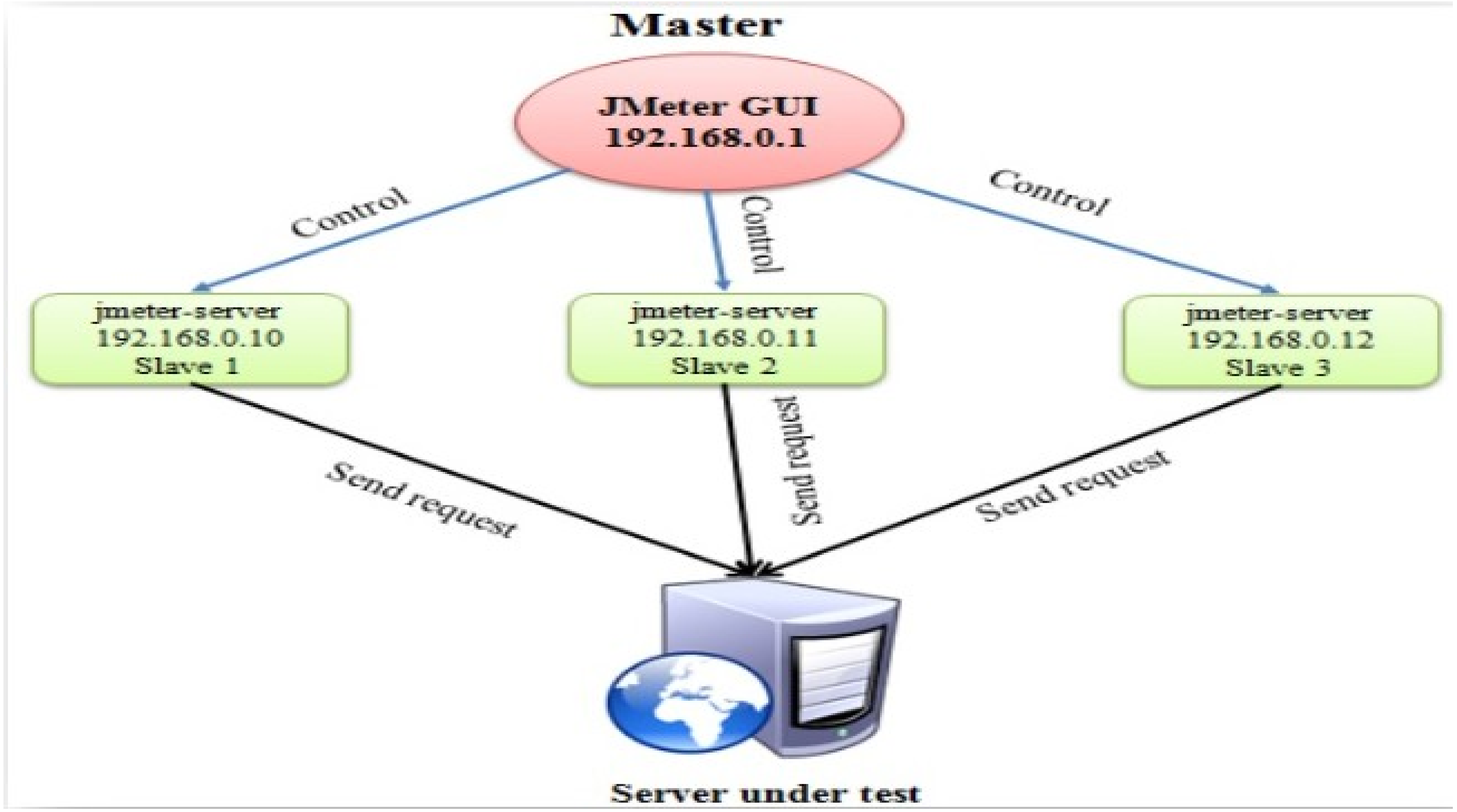
- Extract the zip file and navigate to JMeter→ bin→ double click on windows batch file

- Run the Jmeter.sh for Linux

2. Get Started

Setting up Environment

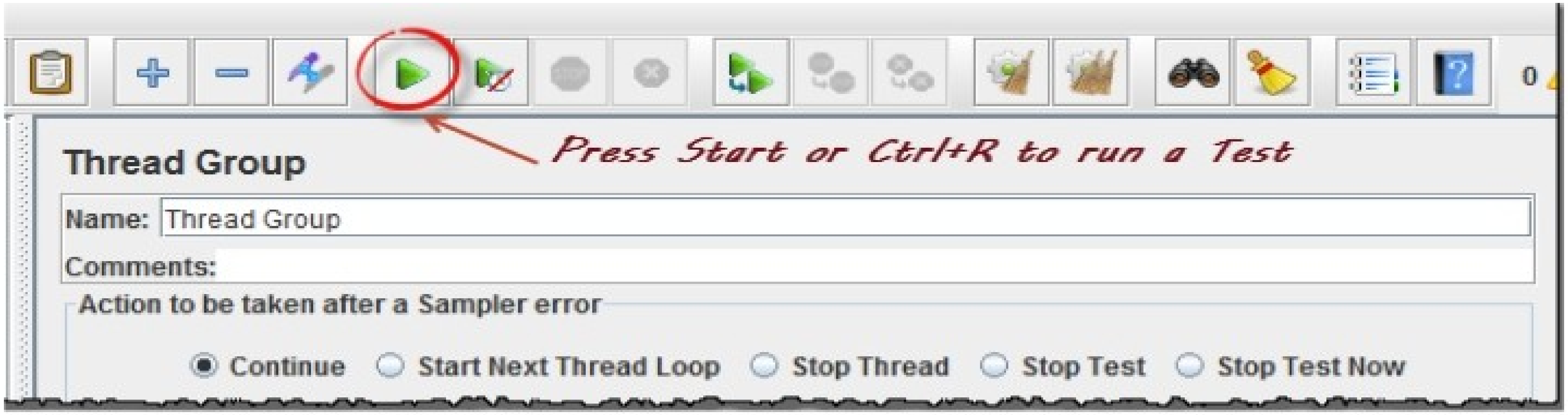
JMeter have simple environment setup



2. Get Started

Running JMeter

- ❖ Open command prompt (use administrative mode to avoid unnecessary hassle)
- ❖Traverse to [Jmeter installation path]\ bin
- ❖ Run Jmeter.bat file or Jmeter.sh



Press Start or Ctrl+R to run a Test

Thread Group

Name: Thread Group

Comments:

Action to be taken after a Sampler error

[✓ Continue] Start Next Thread Loop Stop Thread Stop Test Stop Test Now

3. Introduction to Elements of JMeter Test Plan

Contents

- Test Plan
- Thread Group
- Controllers
- Samplers
- Logic Controllers
- Listeners & Timers
- Assertions
- Configuration Elements
- Pre &Post-Processor Elements

3. Introduction to Elements of JMeter Test Plan

Test Plan

- Test plan will contain the actual test
- It’s a layout of how and what to test
- Test plan describes a series of steps Jmeter will execute once the test plan runs
- A Test plan must contain at least one Thread Group

3. Introduction to Elements of JMeter Test Plan

Thread Group

- Thread group is used for representing users

- There could be one or more Thread Groups in a Test Plan

- Each thread group will execute completely independently from each other

- Thread group can control
  - ➢ Number of users simulated
  - ➢ Ramp up time(How long it will take to start all the threads)
  - ➢ Number of times to perform the test

3. Introduction to Elements of JMeter Test Plan

Controllers

- Controllers are used for grouping and applying logic to test items

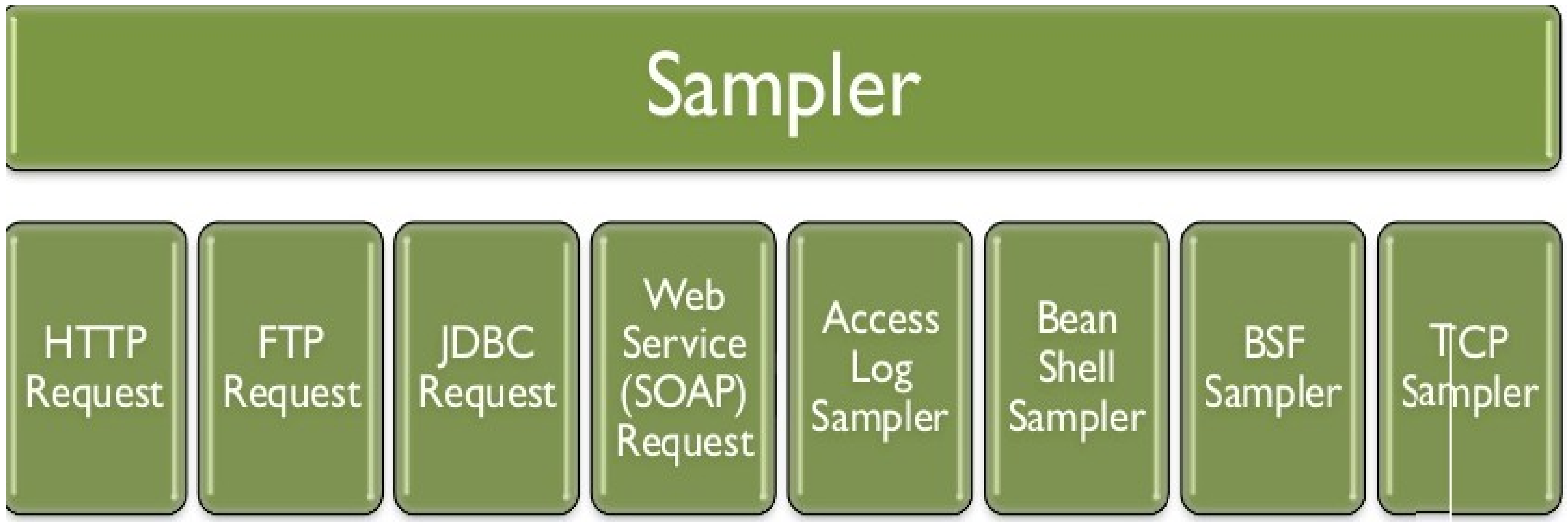
- We have two types of controllers

➢ Sampler - will send requests to the server
➢ Logical Controller - customize the logic while sending the request

3. Introduction to Elements of JMeter Test Plan

Samplers

- Used for performing the actual task
- It allows JMeter to send specific types requests to the server
- List of samplers in JMeter

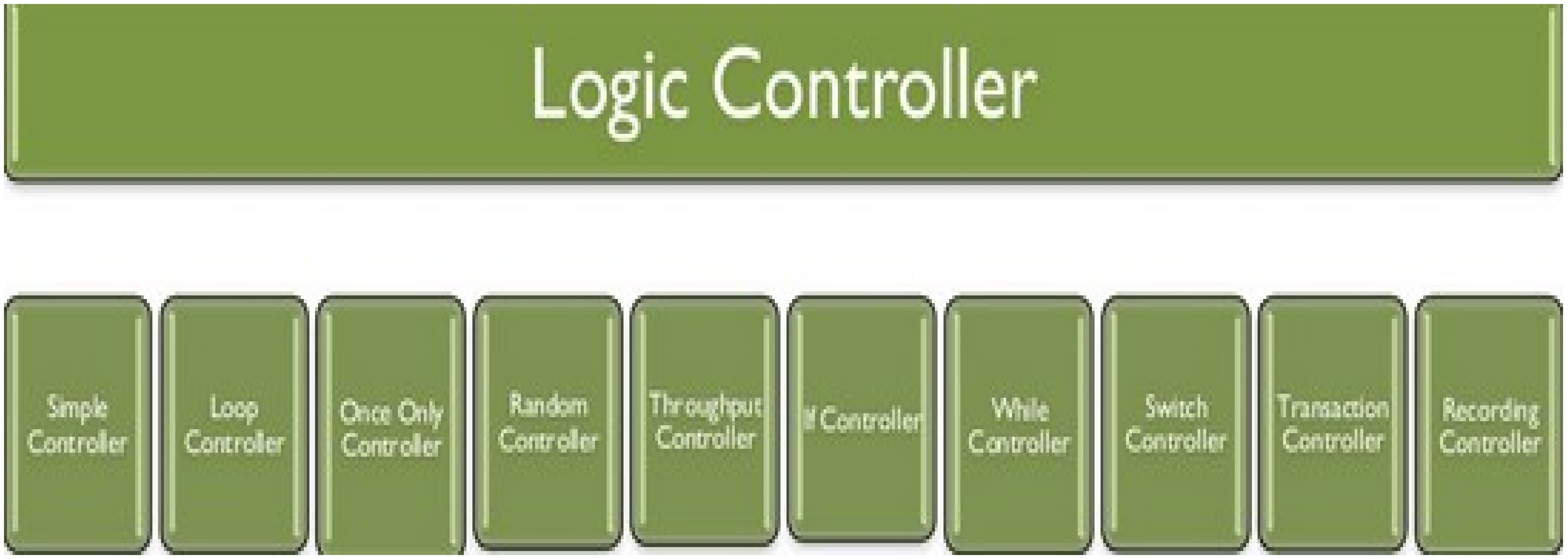


3. Introduction to Elements of JMeter Test Plan

Logic Controllers

- These allow us to customize the logic the JMeter will decide when to send the request

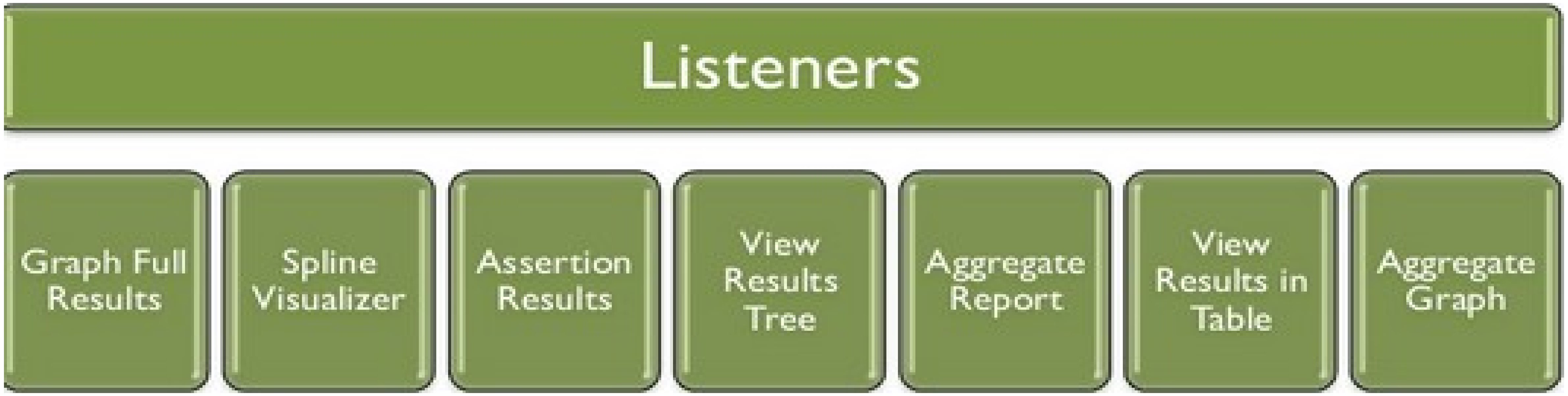
- List of logic controllers in JMeter



3. Introduction to Elements of JMeter Test Plan

Listeners

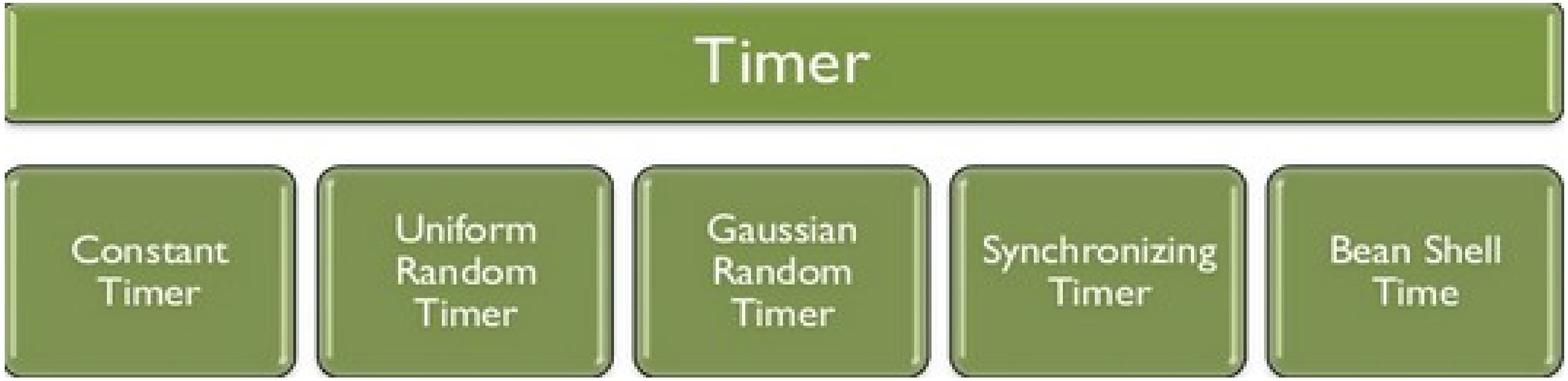
- Listeners are used to displaying the data
- We can view, save the test results
- The following list contains the different types of Listeners available in JMeter



3. Introduction to Elements of JMeter Test Plan

Timers

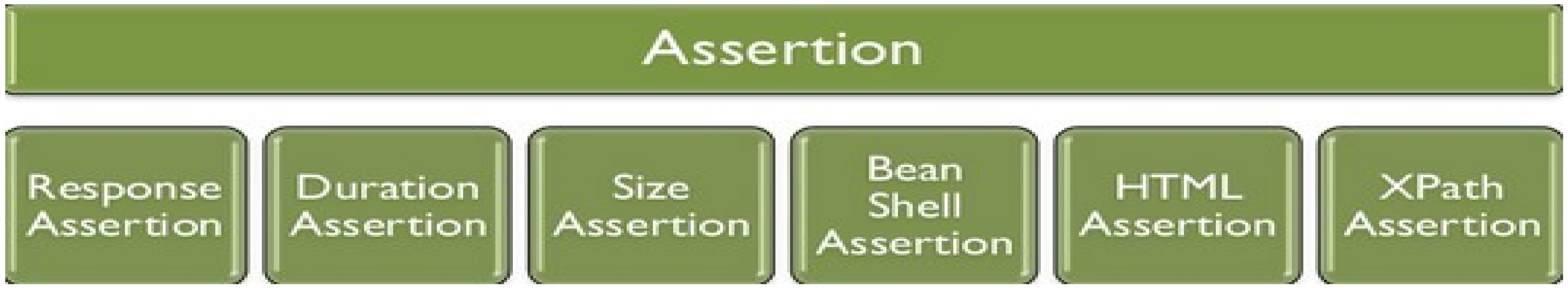
- Allow JMeter to delay between each request that a thread makes
- Think time is used for emulating real life scenarios
- The following list contains the different types of timers available in JMeter



3. Introduction to Elements of JMeter Test Plan

Assertions

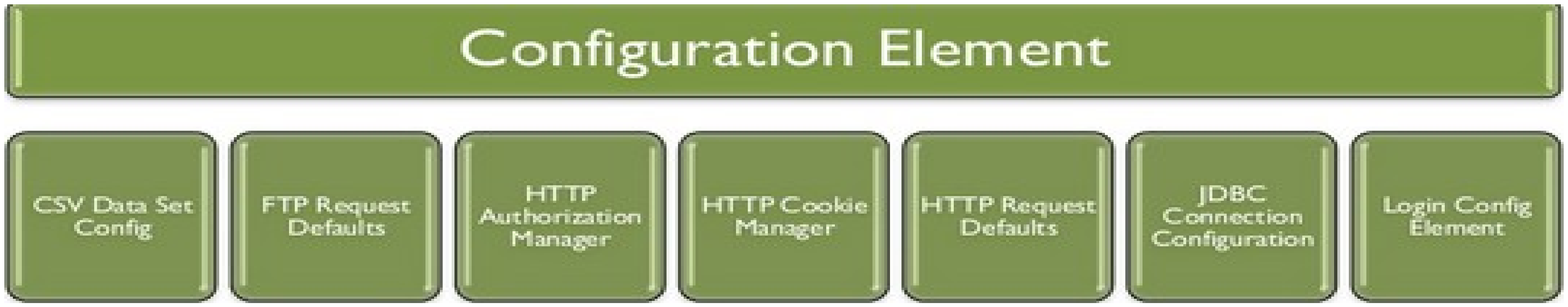
- Used for validating test
- Allow user to test that the application is returning the results you expect it to
- The following list contains the different types of Assertions available in JMeter



3. Introduction to Elements of JMeter Test Plan

Configuration Elements

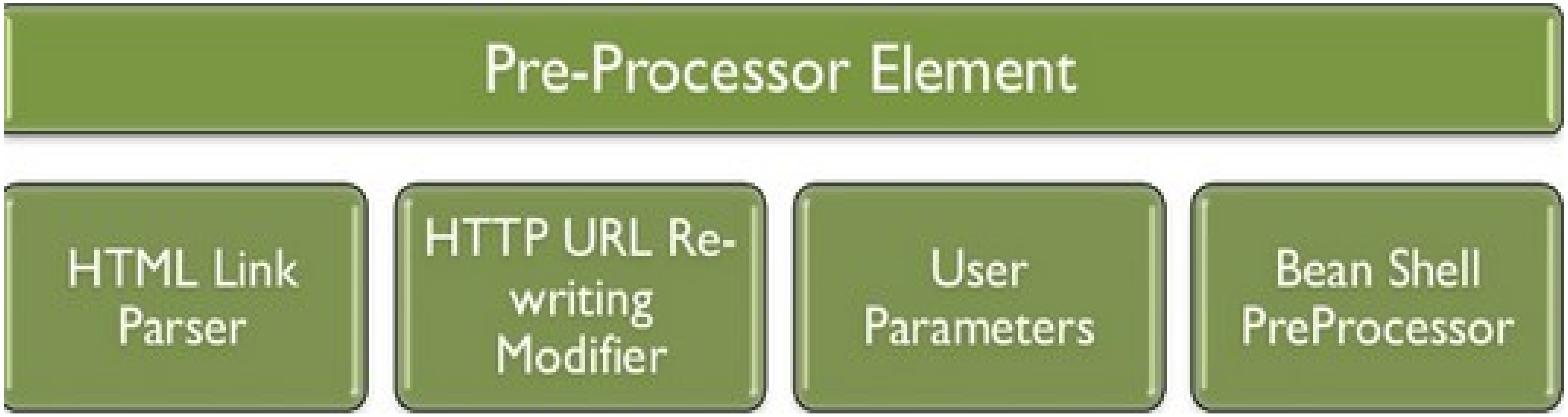
- They are used to add or modify request made by samplers
- They works closely with Sampler request just before it runs
- Below is the list of Configuration elements in JMeter



3. Introduction to Elements of JMeter Test Plan

Pre-Processor Elements

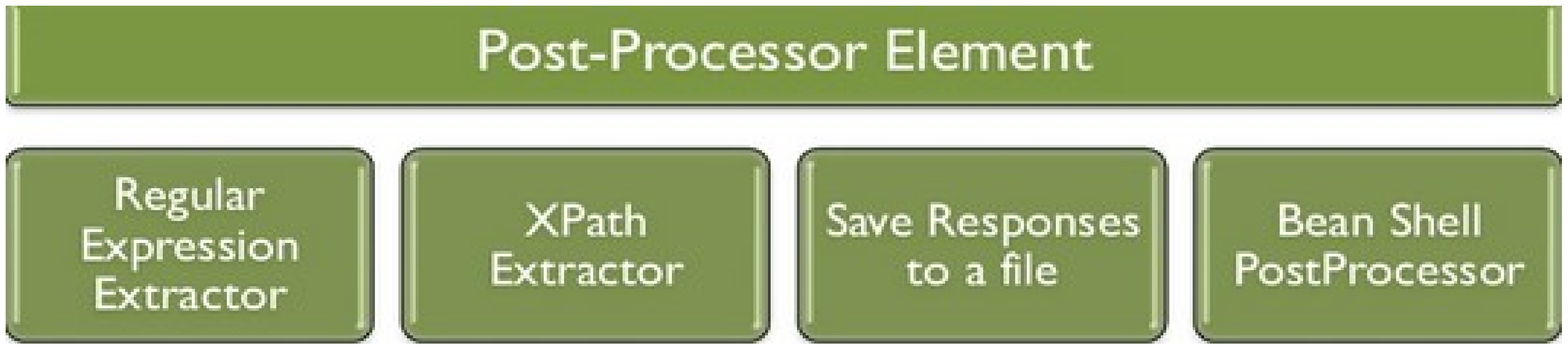
- They execute prior to sampler requests
- They are used to modify the settings of a sampler request just before it runs
- Below is the list of Pre-Processor elements in JMeter



3. Introduction to Elements of JMeter Test Plan

Post-Processor Elements

- They execute some action after sampler request
- It Executes after a request has been made from a sampler
- Below is the list of Post-Processor elements in JMeter



4. Building a Web Test Plan

Contents

4. Building a Web Test Plan

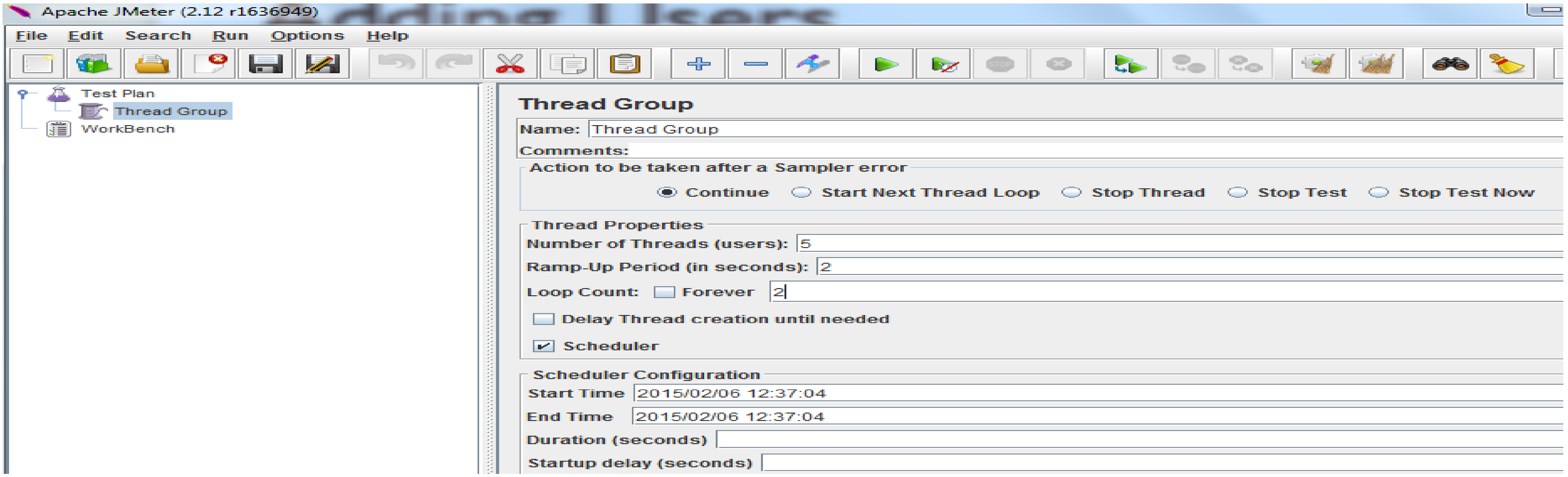
Recording and Playback

- JMeter will act as a proxy server between the browser and the web recording actions
- This will help in writing the web tests
- To create the scripts we can use
  - ➢ Proxy server
  - ➢ Script recording tool(Badboy, Wireshark)

4. Building a Web Test Plan

Adding Users

- 5 users send 2 requests on Jmeter and repeat it twice(5 users x 2 requests x 2 repeat = 20 requests)
- Right click on the test note >> add >> Thread >> Thread Group



Thread Group

Name: Thread Group

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): 5

Ramp-Up Period (in seconds): 2

Loop Count: Forever 2

Delay Thread creation until needed

[✓ Scheduler]

Scheduler Configuration

Start Time 2015/02/06 12:37:04

End Time 2015/02/06 12:37:04

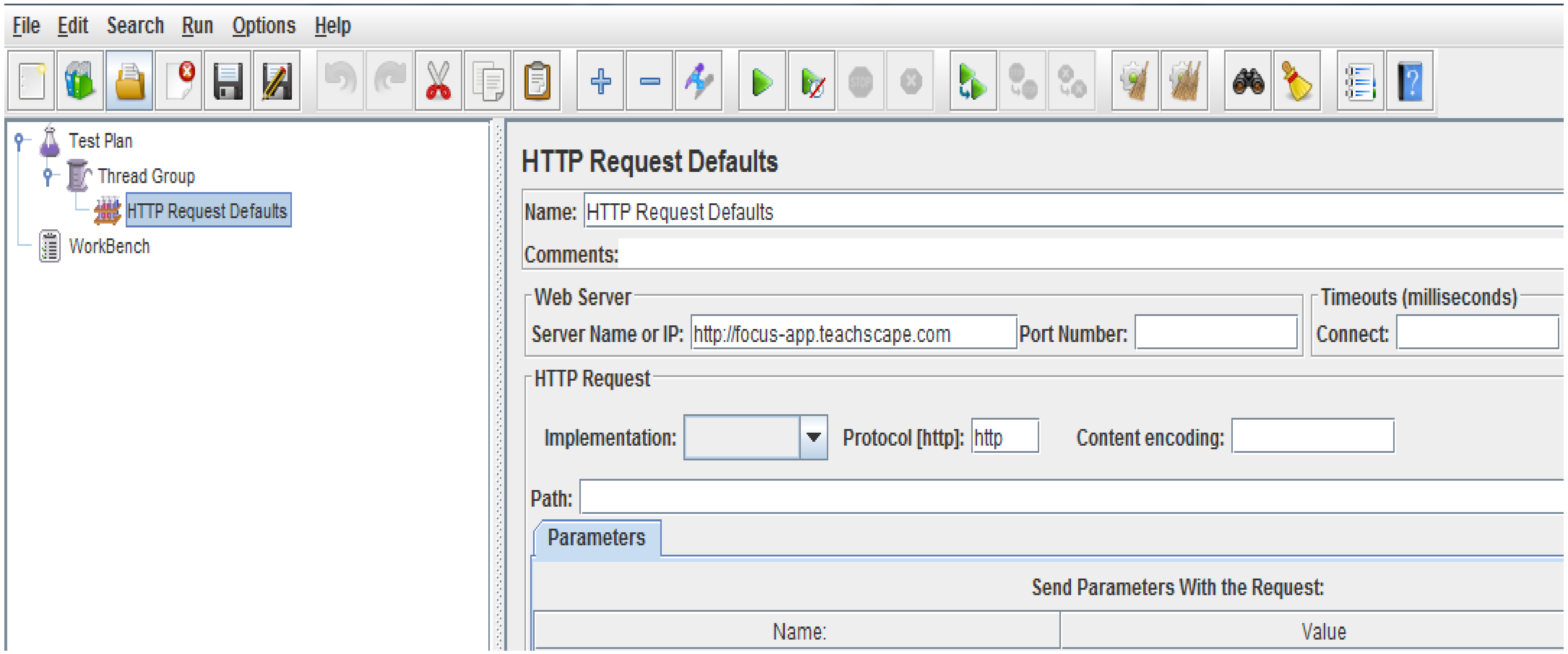
Duration (seconds)

Startup delay (seconds)

4. Building a Web Test Plan

Adding default HTTP Request Properties

- This will let us define default http parameter for every request
- We can add this from Add >>Config Element >> HTTP request default



HTTP Request Defaults

Name: HTTP Request Defaults

Comments:

Web Server

Server Name or IP: http://focus-app.teachscape.com

Port Number:

Timeouts (milliseconds)

Connect:

HTTP Request

Implementation:

Protocol [http]: http

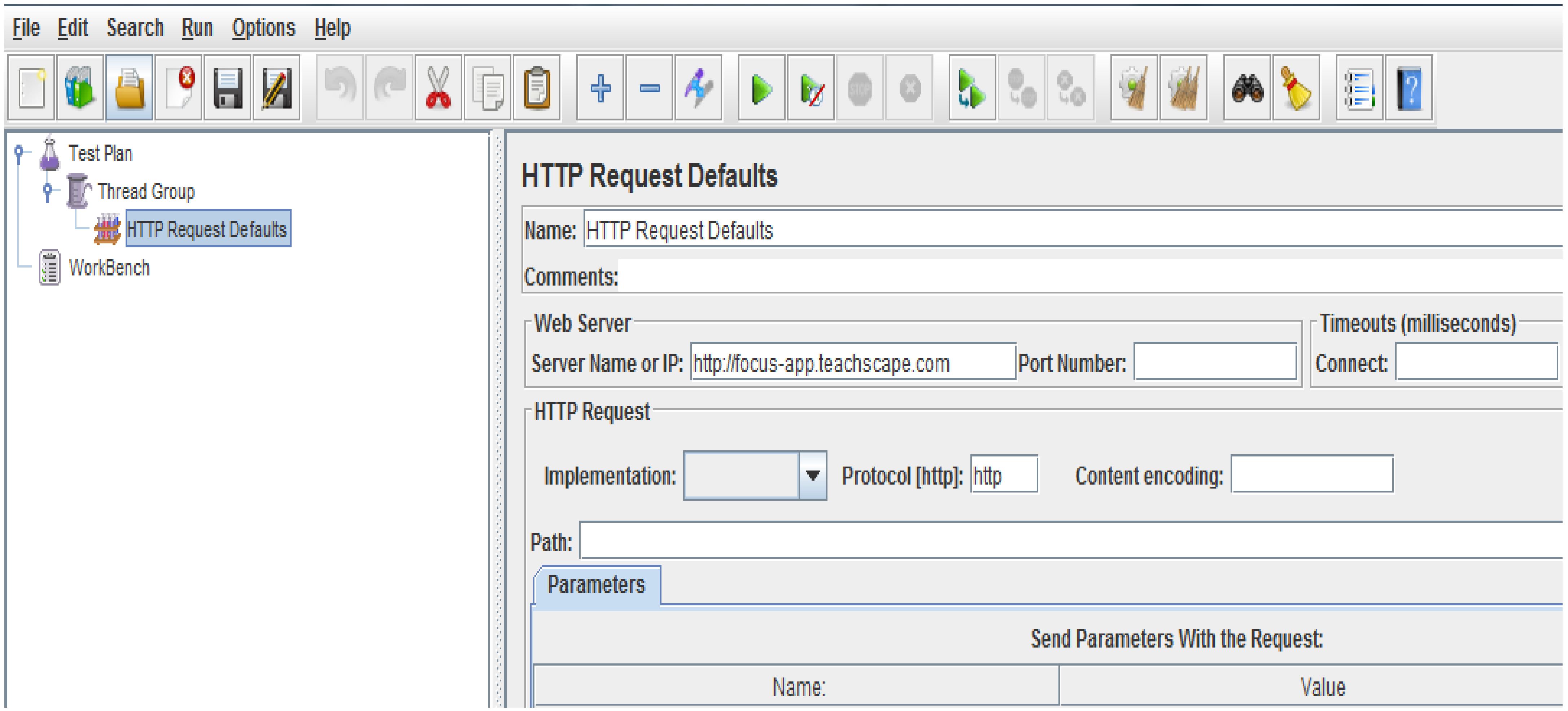
Content encoding:

Path:

4. Building a Web Test Plan

Adding HTTP Request

- We can add this from Add >> Sampler >> HTTP Request



HTTP Request Defaults

Name: HTTP Request Defaults

Comments:

Web Server

Server Name or IP: http://focus-app.teachscape.com

Port Number:

Timeouts (milliseconds)

Connect:

HTTP Request

Implementation:

Protocol [http]: http

Content encoding:

Path:

Parameters

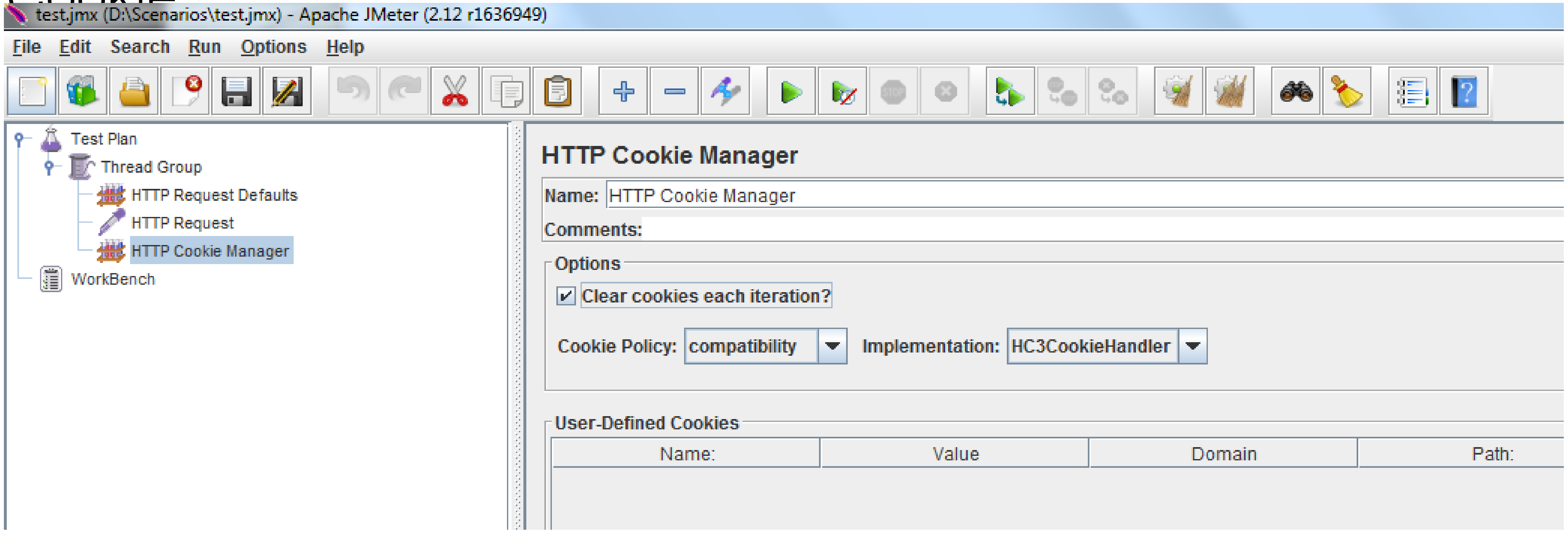
Send Parameters With the Request:

Name:	Value

4. Building a Web Test Plan

Adding Cookie Support

- They are normally included to maintain a certain state for each user
- We can add this from Add >> Config Element >> HTTP Cookie


test.jmx (D:\Scenarios\test.jmx) - Apache JMeter (2.12 r1636949)

HTTP Cookie Manager

Name: HTTP Cookie Manager

Comments:

Options

[✓] Clear cookies each iteration?

Cookie Policy: compatibility Implementation: HC3CookieHandler

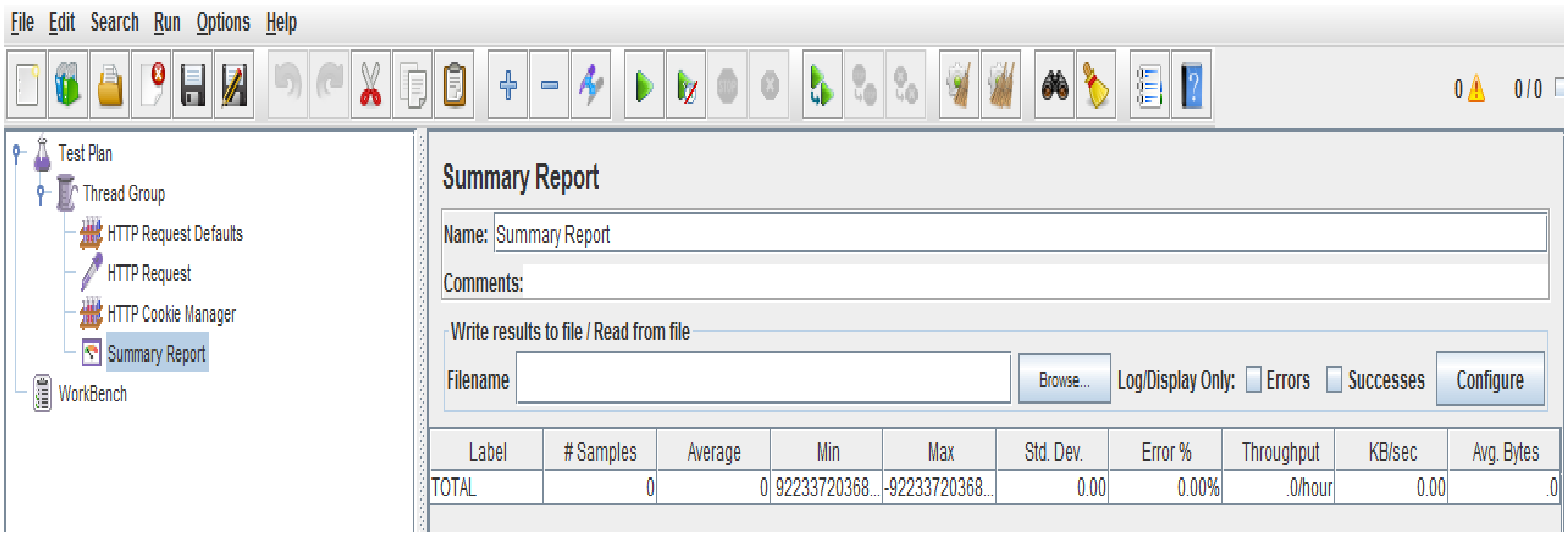
User-Defined Cookies

Name:	Value	Domain	Path:

4. Building a Web Test Plan

Adding a listener to view or store results

- Listener responsible for storing all the results of our HTTP request and presenting in Visualize mode
- You can add it from Add >> Listener>> Summary report



Test Plan
Thread Group
HTTP Request Defaults
HTTP Request
HTTP Cookie Manager
Summary Report
WorkBench

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename

Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
TOTAL	0	0	92233720368...	-92233720368...	0.00	0.00%	.0/hour	0.00	.0

5. Load & Performance testing of a web site

Contents

5. Load & Performance testing of a web site

Preparing for Load Testing

❖ Need to address a number of concerns with regards to the target server under test.

❖ A load test helps to benchmark performance behavior of a server, it is important to be able to identify the expectations and other matters that would normally be taken into account in order to carry out a successful load testing

5. Load & Performance testing of a web site

Need to Know

- A suitable time to load – test the application, for instance when no development work is taken place on the server and / or no other users are accessing the server
- The performance metrics, accepted levels, or SLA's and goals.
- Objective of the test.
- The internet protocol(s) the application is (are) using (Http, Https.ftp, etc.)
- If your application has a state, the method used to manage it (URL rewriting, cookies, etc.)
- The workload at normal time and peak time.

5. Load & Performance testing of a web site

Some Helpful Tips to Get Better Results

- ✓ Use meaningful test scenarios to construct real –life test cases.
- ✓ Run JMeter on a machine other than that running the application.
- ✓ The machine running JMeter should have sufficient network bandwidth, memory , CPU to generate load.
- ✓ Let JMeter test plan run for long period , hours and days, or for a large number of iterations.
- ✓ Ensure that the application should be stable and optimized for one user before testing it for concurrent users.
- ✓ Incorporate think time or delay using timers in your JMeter test plan
- ✓ Keep close watch on the four main things: Processer , memory, disk and network.

5. Load & Performance testing of a web site

Using JMeter Components

❖ We will test five key scenario’s:-

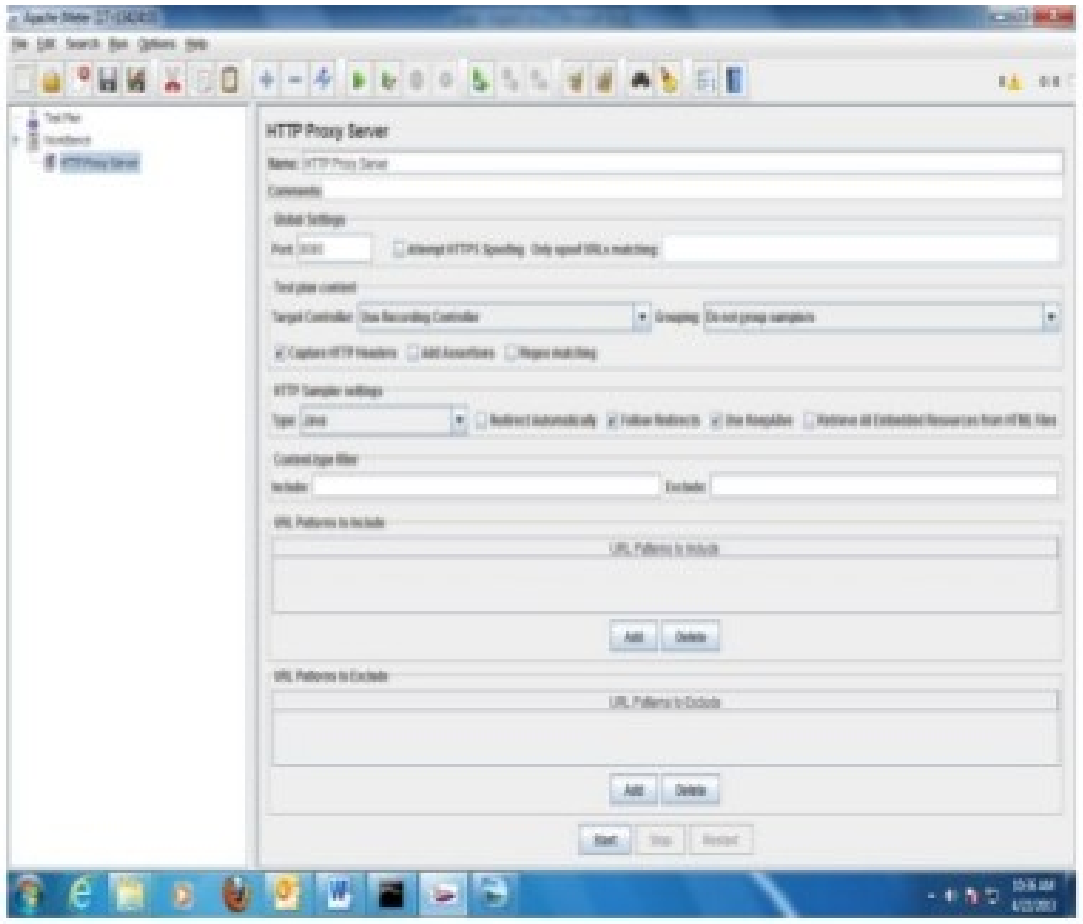
- Home page
- Keyword search
- Create account
- Select a title
- Add to cart

These scenarios will be included in a single JMeter test plan, for a simplicity reasons.

5. Load & Performance testing of a web site

Recording HTTP Requests

- ❖ JMeter Proxy can use to record all request sent to server.
- ❖ Create test plan with default http testing.
- ❖ Add HTTP proxy server under work bench node
- ❖ Define the port number of proxy server.



HTTP Proxy Server

Name: HTTP Proxy Server

Comments:

Global Settings

Test plan content

Target Controller: Use Recording Controller

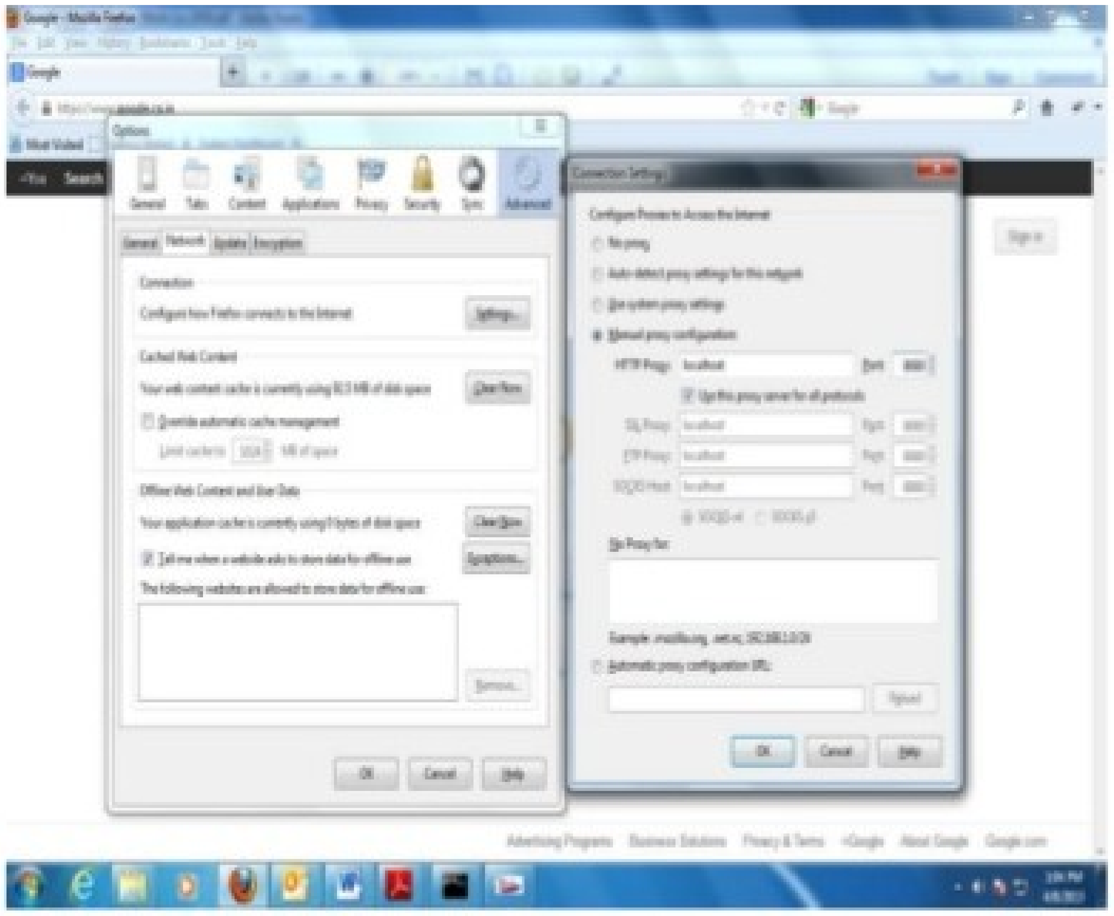
HTTP Sampler settings

Content-type filter

Include: Exclude:

URL Patterns to Include

URL Patterns to Exclude



5. Load & Performance testing of a web site

Creating the Test Plan

- Right click on the test plan elements and select Add/ thread group.
- Configure the number of threads to 10, Ramp-up period to 1 second and

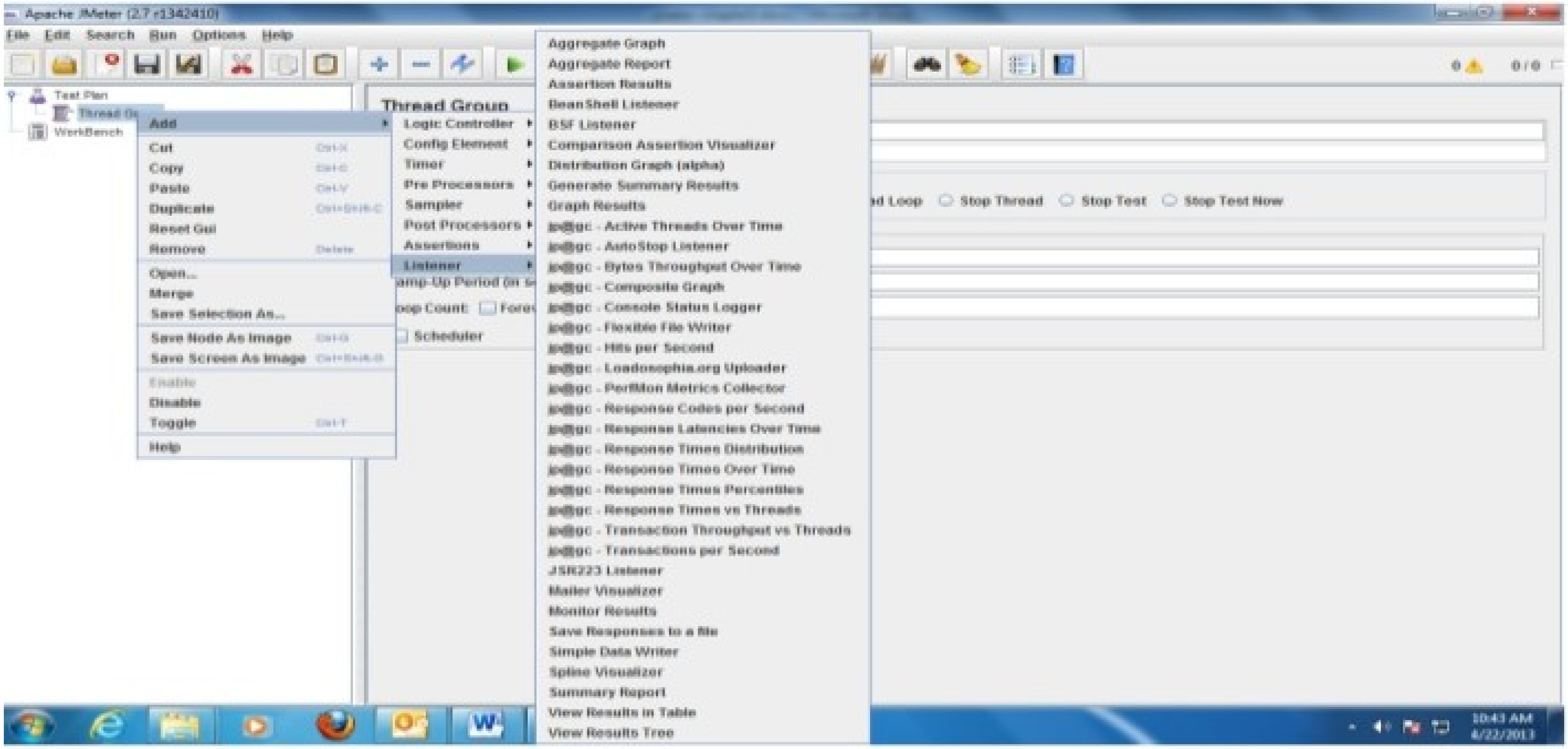
  loop count to 50
- Add to the test plan config element / HTTP Request defaults.
- Add pre- Processer/ user parameter element or config element / csv data

  set config.

5. Load & Performance testing of a web site

Adding Listeners

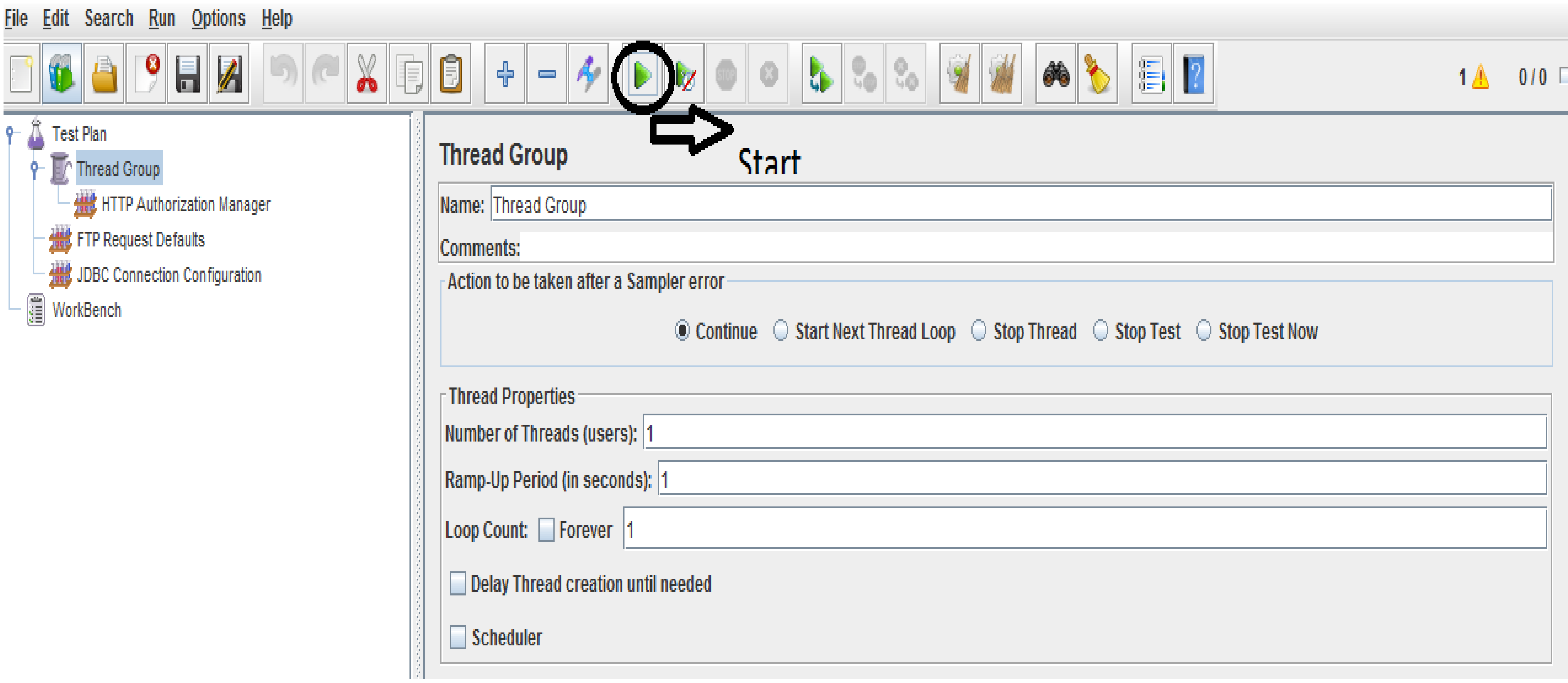
❖ Right click thread group name >> Add>> Listener.


Apache JMeter (2.7 r1342410)
File Edit Search Run Options Help
Test Plan
Thread Gr
WorkBench
Add
Cut Ctrl-X
Copy Ctrl-C
Paste Ctrl-V
Duplicate Ctrl+Shift-C
Reset Gui
Remove Delete
Open...
Merge
Save Selection As...
Save Node As Image Ctrl-G
Save Screen As Image Ctrl+Shift-G
Enable
Disable
Toggle Ctrl-T
Help
Thread Group
Logic Controller
Config Element
Timer
Pre Processors
Sampler
Post Processors
Assertions
Listener
Aggregate Graph
Aggregate Report
Assertion Results
BeanShell Listener
BSF Listener
Comparison Assertion Visualizer
Distribution Graph (alpha)
Generate Summary Results
Graph Results
jp@gc - Active Threads Over Time
jp@gc - AutoStop Listener
jp@gc - Bytes Throughput Over Time
jp@gc - Composite Graph
jp@gc - Console Status Logger
jp@gc - Flexible File Writer
jp@gc - Hits per Second
jp@gc - Loadosophia.org Uploader
jp@gc - PerfMon Metrics Collector
jp@gc - Response Codes per Second
jp@gc - Response Latencies Over Time
jp@gc - Response Times Distribution
jp@gc - Response Times Over Time
jp@gc - Response Times Percentiles
jp@gc - Response Times vs Threads
jp@gc - Transaction Throughput vs Threads
jp@gc - Transactions per Second
JSR223 Listener
Mailer Visualizer
Monitor Results
Save Responses to a file
Simple Data Writer
Spline Visualizer
Summary Report
View Results in Table
View Results Tree
Stop Thread
Stop Test
Stop Test Now
Scheduler
10:43 AM
4/22/2013

5. Load & Performance testing of a web site

Running the Test Plan

- When JMeter Scripts get ready .
- Click on “Start.”



Thread Group

Name: Thread Group

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): 1

Ramp-Up Period (in seconds): 1

Loop Count: ☐ Forever 1

☐ Delay Thread creation until needed

☐ Scheduler

5. Load & Performance testing of a web site

Interpreting the Results

- Once the test is completed, we can now retrieve the results we have save for each controller.
- With the exception of Assertion Result Listener, the saved data can be viewed in numerous forms.

5. Load & Performance testing of a web site

Monitoring the Server's Performance

- There is a special Listener that allows you to monitor the server target server’s performance as Sampler make request.
- This monitor Result Listener is designed to work with Apache tomcat version 5 and above.
- Some server monitor tools.
    - Perfmon.
    - Blaze meter.

5. Load & Performance testing of a web site

Performance Test Reporting

- ❖ Introduction
- ❖ Test Environment (software and hardware)
- ❖ Goals of this Report.
- ❖ Assumption.
- ❖ Performance Requirements.
- ❖ Workload Scenario.
- ❖ Performance testing approach.
- ❖ Results.
- ❖ Samples response monitoring(Jmeter Results)
  Run Date: 2/09/2015
  Duration: 1 hour, 40 min, 40 sec.
  Ramp up increment: 35 users
  Think time: varies
  Connection Speed: BroadBand

6. Parameterize with test data

Contents

- What is Parameterization
- Why Parameterize
- Identifying the test data on AUT
- Using the CSV Data Config in JMeter Tests

6. Parameterize with test data

What is Parameterization

- It is the way of replacing a hardcoded value in the script with a parameter which represents a list of values.
- It is a script that contains the actual values used during recording and during script enhancement phase test engineer has to replace the recorded values with parameters is known as parameter-zing the script.

6. Parameterize with test data

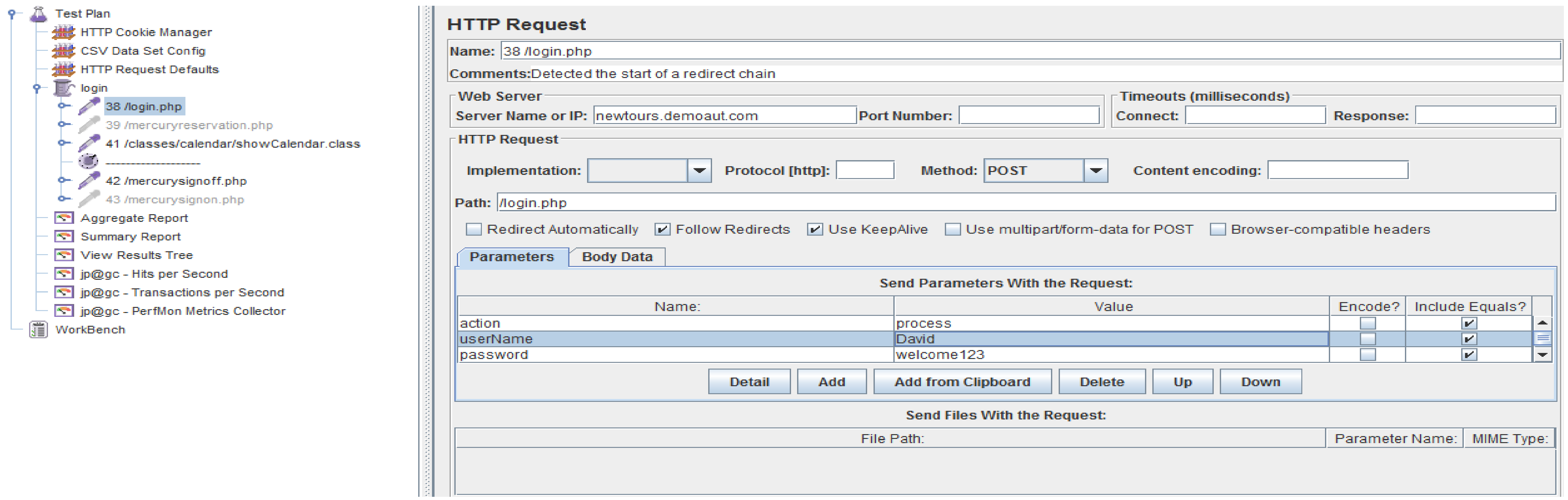
Why Parameterize

- It allows us to test your script with different values
- Replace the constant values in script with parameters
- Simulate real user behavior while running the rest

6. Parameterize with test data

Identifying the test data on AUT

- While login “User name” & “Password” are the parameters for which test data needed
- While registration “First name” , “City” will be the parameters for which test data is needed



Test Plan
- HTTP Cookie Manager
- CSV Data Set Config
- HTTP Request Defaults
- login
  - 38 /login.php
  - 39 /mercuryreservation.php
  - 41 /classes/calendar/showCalendar.class
  - ------------------
  - 42 /mercurysignoff.php
  - 43 /mercurysignon.php
- Aggregate Report
- Summary Report
- View Results Tree
- jp@gc - Hits per Second
- jp@gc - Transactions per Second
- jp@gc - PerfMon Metrics Collector

WorkBench

HTTP Request

Name: 38 /login.php

Comments: Detected the start of a redirect chain

Web Server
Server Name or IP: newtours.demoaut.com **Port Number:**

Timeouts (milliseconds)
Connect: **Response:**

HTTP Request
Implementation: **Protocol [http]:** **Method:** POST **Content encoding:**

Path: /login.php

☐ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive ☐ Use multipart/form-data for POST ☐ Browser-compatible headers

Parameters | Body Data

Send Parameters With the Request:

Name:	Value	Encode?	Include Equals?
action	process	☐	☑
userName	David	☐	☑
password	welcome123	☐	☑

Detail | Add | Add from Clipboard | Delete | Up | Down

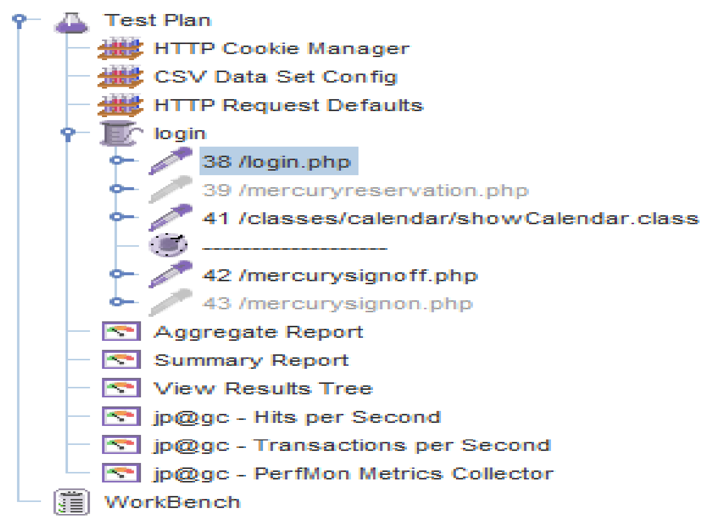
Send Files With the Request:

File Path:	Parameter Name:	MIME Type:

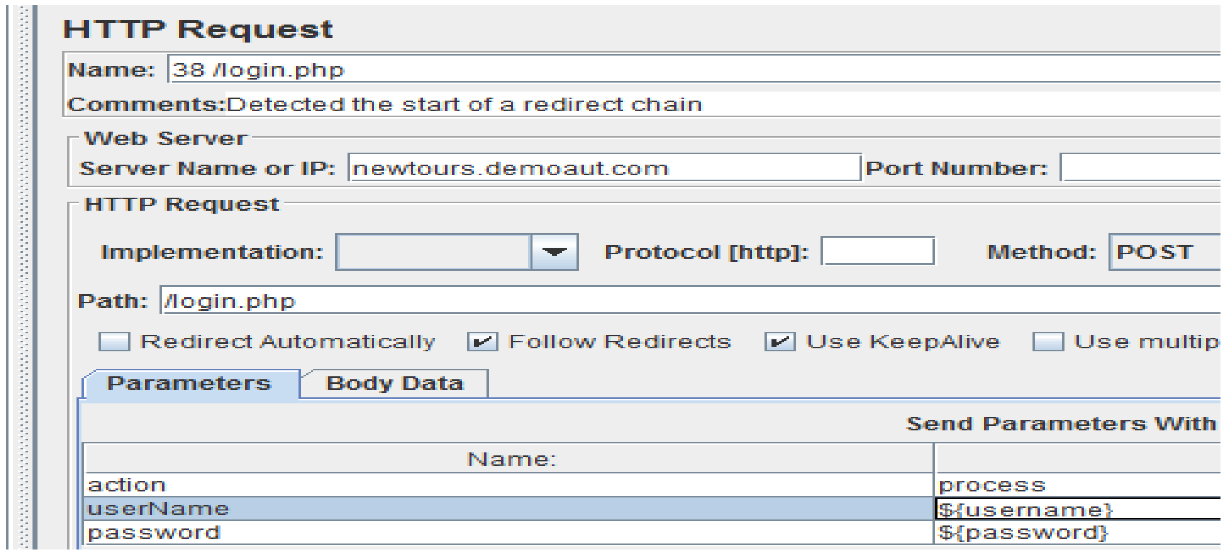
6. Parameterize with test data

Using CSV Data Config in Jmeter Tests

- Create a CSV file with the list of username and password
- Store in the same folder where we store the tests
- Add CSV data set into the test tree from config elements
- Add ${username},${password}in request sampler as parameter



```
Test Plan
    HTTP Cookie Manager
    CSV Data Set Config
    HTTP Request Defaults
    login
        38 /login.php
        39 /mercuryreservation.php
        41 /classes/calendar/showCalendar.class
        ----------------------
        42 /mercurysignoff.php
        43 /mercurysignon.php
    Aggregate Report
    Summary Report
    View Results Tree
    jp@gc - Hits per Second
    jp@gc - Transactions per Second
    jp@gc - PerfMon Metrics Collector
WorkBench
```



HTTP Request

Name: 38 /login.php

Comments:Detected the start of a redirect chain

Web Server

Server Name or IP: newtours.demoaut.com Port Number:

HTTP Request

Implementation: Protocol [http]: Method: POST

Path: /login.php

☐ Redirect Automatically ✓ Follow Redirects ✓ Use KeepAlive ☐ Use multip

Parameters | Body Data

Send Parameters With

Name:	
action	process
userName	${username}
password	${password}

7. Adding Assertions to the test script

Contents

- What is Assertion
- Why Assertion
- Types of Assertion in JMeter
- Running the tests and analyzing the Assertion

7. Adding Assertions to the test script

What is Assertion

- Assertion allow you to include some validation test on the response

- To ensure the response you receive.

7. Adding Assertions to the test script

Why Assertion

- Assertion are used to perform additional checks on samplers.
- They are processed after every sampler in the same scope.

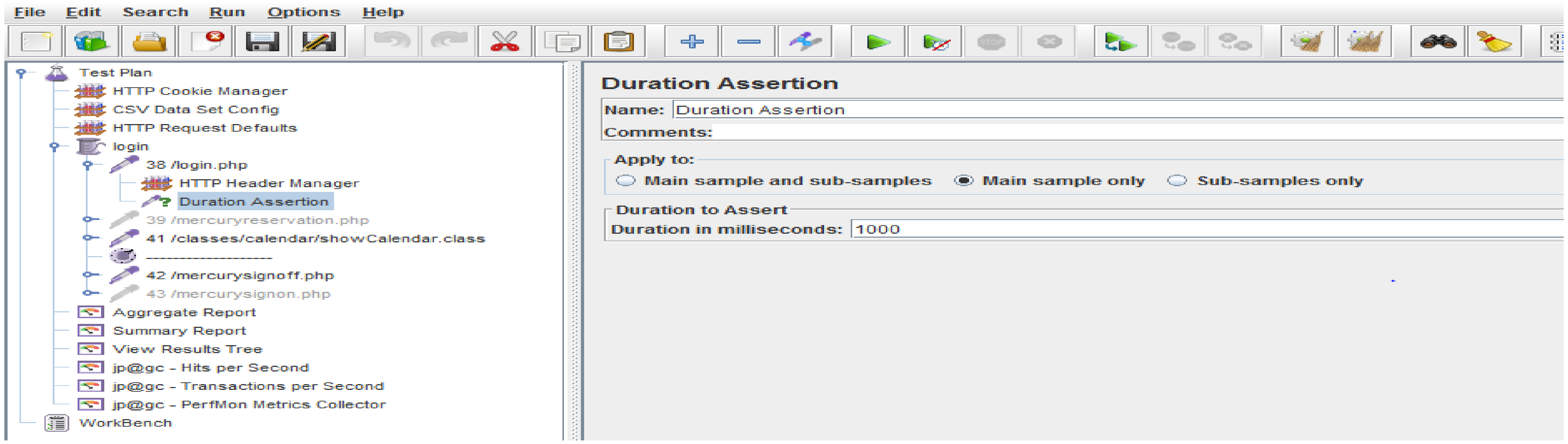
7. Adding Assertions to the test script

Types of Assertion in Jmeter

Repsonse Assertion: Compared against various fields of response
Duration Assetrion: Tests each response was received with a given amount of time
Size Assertion: Tests each response contains the right number of bytes in it
XML Assertion: Tests response data consists of a formally correct XML document
Beanshell Assertion: Checking using a Beanshell script
MD5Hex Assertion:Check the MD5 hash of the response data
HTML Assertion:Check HTML syntax of the response
Xpath Assertion:Checks the document for well formedness
XML Schema Assertion:Allows the user to validate a response against an XML Schema

7. Adding Assertions to the test script

Running the test and analyzing the Assertion results



```
Test Plan
    HTTP Cookie Manager
    CSV Data Set Config
    HTTP Request Defaults
    login
        38 /login.php
            HTTP Header Manager
            Duration Assertion
        39 /mercuryreservation.php
        41 /classes/calendar/showCalendar.class
        -------------------
        42 /mercurysignoff.php
        43 /mercurysignon.php
    Aggregate Report
    Summary Report
    View Results Tree
    jp@gc - Hits per Second
    jp@gc - Transactions per Second
    jp@gc - PerfMon Metrics Collector
WorkBench
```

Duration Assertion

Name: Duration Assertion

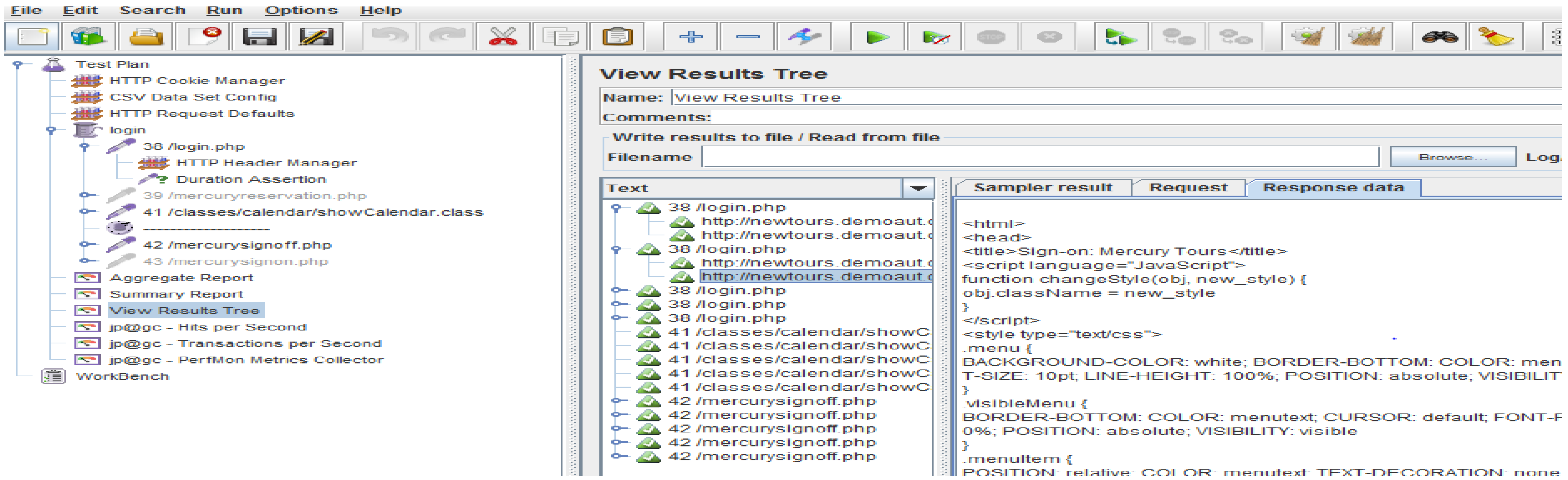
Comments:

Apply to:

( ) Main sample and sub-samples [✓ Main sample only] ( ) Sub-samples only

Duration to Assert

Duration in milliseconds: 1000



```
Test Plan
    HTTP Cookie Manager
    CSV Data Set Config
    HTTP Request Defaults
    login
        38 /login.php
            HTTP Header Manager
            Duration Assertion
        39 /mercuryreservation.php
        41 /classes/calendar/showCalendar.class
        -------------------
        42 /mercurysignoff.php
        43 /mercurysignon.php
    Aggregate Report
    Summary Report
    View Results Tree
    jp@gc - Hits per Second
    jp@gc - Transactions per Second
    jp@gc - PerfMon Metrics Collector
WorkBench
```

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename Browse... Log

Text

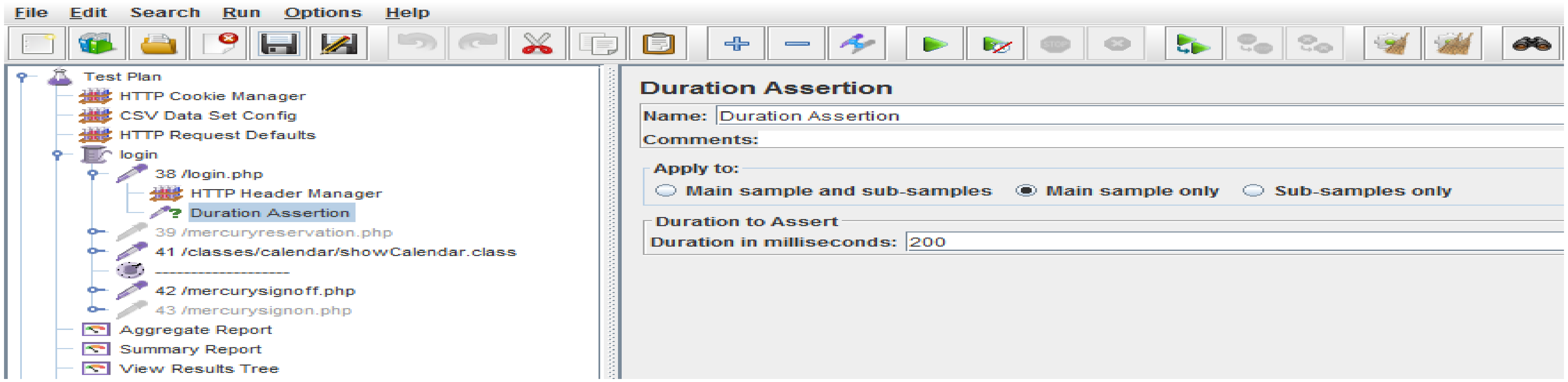
```
38 /login.php
    http://newtours.demoaut.
    http://newtours.demoaut.
38 /login.php
    http://newtours.demoaut.
    http://newtours.demoaut.
38 /login.php
38 /login.php
38 /login.php
41 /classes/calendar/showC
41 /classes/calendar/showC
41 /classes/calendar/showC
41 /classes/calendar/showC
41 /classes/calendar/showC
42 /mercurysignoff.php
42 /mercurysignoff.php
42 /mercurysignoff.php
42 /mercurysignoff.php
42 /mercurysignoff.php
```

Sampler result | Request | Response data

```
<html>
<head>
<title>Sign-on: Mercury Tours</title>
<script language="JavaScript">
function changeStyle(obj, new_style) {
obj.className = new_style
}
</script>
<style type="text/css">
.menu {
BACKGROUND-COLOR: white; BORDER-BOTTOM: COLOR: men
T-SIZE: 10pt; LINE-HEIGHT: 100%; POSITION: absolute; VISIBILIT
}
.visibleMenu {
BORDER-BOTTOM: COLOR: menutext; CURSOR: default; FONT-F
0%; POSITION: absolute; VISIBILITY: visible
}
.menuItem {
POSITION: relative; COLOR: menutext; TEXT-DECORATION: none
```

7. Adding Assertions to the test script

Running the test and analyzing the Assertion results



Test Plan
- HTTP Cookie Manager
- CSV Data Set Config
- HTTP Request Defaults
- login
  - 38 /login.php
    - HTTP Header Manager
    - Duration Assertion
  - 39 /mercuryreservation.php
  - 41 /classes/calendar/showCalendar.class
  - -------------------
  - 42 /mercurysignoff.php
  - 43 /mercurysignon.php
- Aggregate Report
- Summary Report
- View Results Tree

Duration Assertion

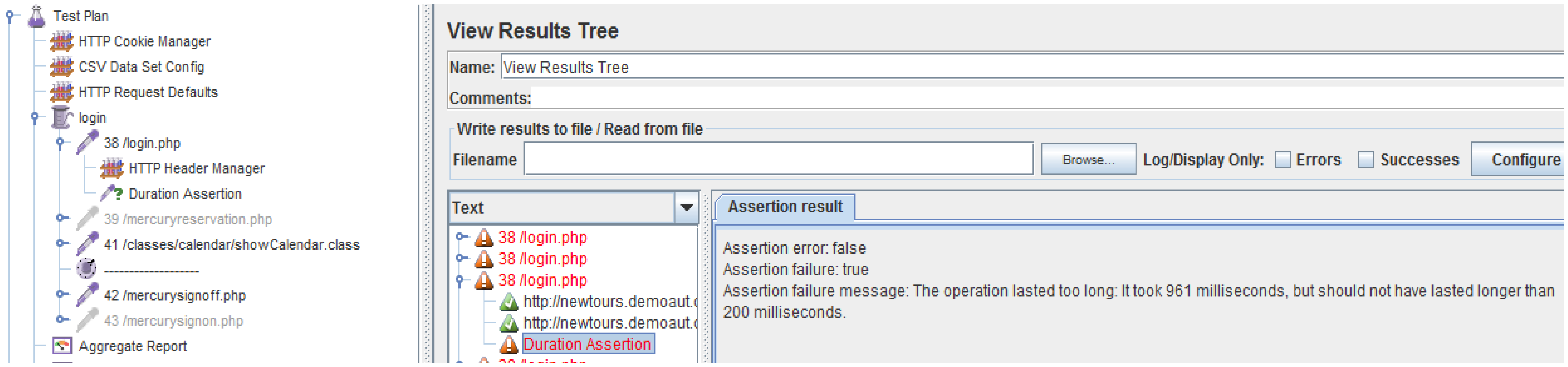
Name: Duration Assertion

Comments:

Apply to:
Main sample and sub-samples [✓ Main sample only] Sub-samples only

Duration to Assert

Duration in milliseconds: 200



Test Plan
- HTTP Cookie Manager
- CSV Data Set Config
- HTTP Request Defaults
- login
  - 38 /login.php
    - HTTP Header Manager
    - Duration Assertion
  - 39 /mercuryreservation.php
  - 41 /classes/calendar/showCalendar.class
  - -------------------
  - 42 /mercurysignoff.php
  - 43 /mercurysignon.php
- Aggregate Report

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename

Log/Display Only: Errors Successes

- 38 /login.php
- 38 /login.php
- 38 /login.php
  - http://newtours.demoaut.
  - http://newtours.demoaut.
  - Duration Assertion

```
Assertion error: false
Assertion failure: true
Assertion failure message: The operation lasted too long: It took 961 milliseconds, but should not have lasted longer than
200 milliseconds.
```

8. Handling the dynamic server values

Contents

8. Handling the dynamic server values

What is Correlation

- A Correlation is a connection or association.

- Capturing the dynamic data that is being generated by the server is called correlation.

- Correlation will be done by regular expression extractor in JMeter.

8. Handling the dynamic server values

Why is Correlation

- To overcome script fail we need a way which can capture these dynamically generated session values and pass it subsequently to any part of the script,
wherever required. This method to identify and set the dynamic generated value is known as correlation.

- Sample of Regular Expression and Usage:
  Session ID = (.+?) -> to correlate the url/dynamic Id.
Ex: Session ID = jkjoujn4348763jh35y9h&OrderID=ikikikii9987kmnhj2

8. Handling the dynamic server values

Regular Expression

- Extract Single string

Suppose we want to match the following portion of a web – page:

name =“file” value=“readme.txt”>

If we want extract readme.txt.. A suitable regular expression will be

name =“file” value=“(.+?)”>

The Special Characters above are

() – enclose the portion of the match string to be returned

. – match any character

+ -- one or more times

? – stop when the first match succeeds

8. Handling the dynamic server values

Regular Expression

- Extract Multiple string

Suppose we want to match the following portion of a web – page:

name =“file” value=“readme.txt”>

If we want extract both file.name and readme.txt..

A suitable regular expression will be

name =“([^”]+)” value=“([^”]+)”>

That would create two groups, which could be used in JMeter Regular expression extractor template as 1 and 2.

For example, assume :

Reference Name : MYREF

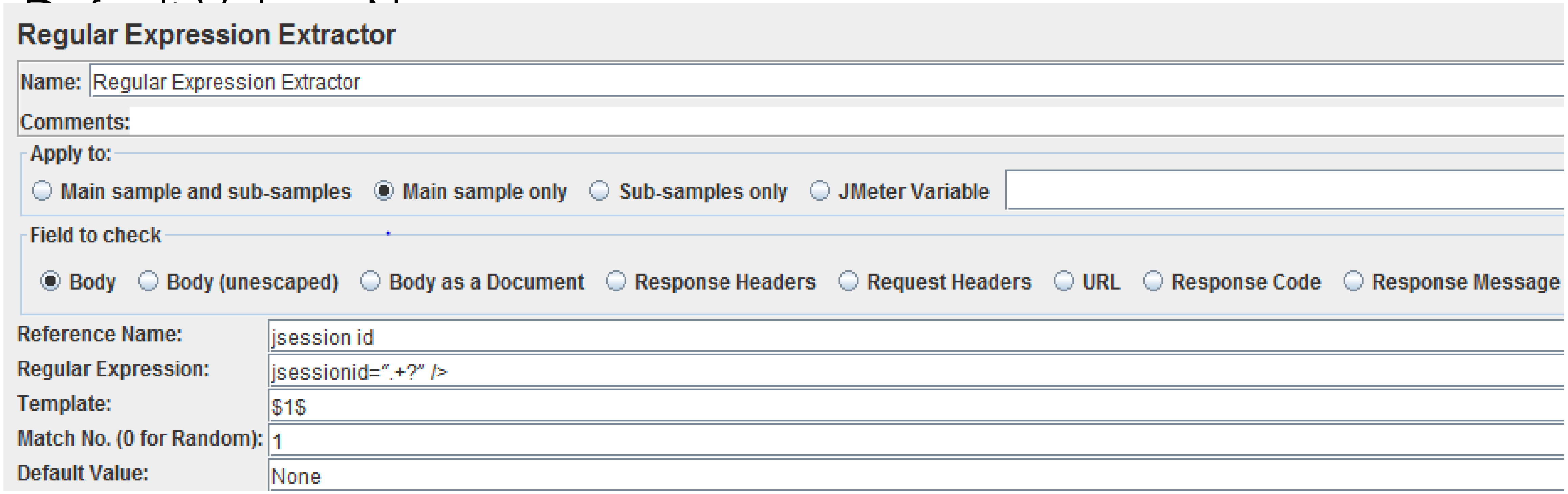
Regex: name=“(.+?)” value=“(.+?)”

Template: $1$$2$

8. Handling the dynamic server values

Using Regular Expression Extractor in Jmeter Tests

Name Regular expression extractor : jsessionid
Reference Name : jsession id
Enter Regular expression : jsessionid = “.+?” />
(.+?) this referes JMeter captures any values in between LB and RB
Template : 1(this if for grouping)
Match No :1
Default Value : N



Regular Expression Extractor

Name: Regular Expression Extractor
Comments:

Apply to:
Main sample and sub-samples [✓ Main sample only] Sub-samples only JMeter Variable

Field to check
[✓ Body] Body (unescaped) Body as a Document Response Headers Request Headers URL Response Code Response Message

Reference Name:	jsession id
Regular Expression:	jsessionid=“.+?” />
Template:	1
Match No. (0 for Random):	1
Default Value:	None

9. Best Practices

Content.

- ❖ Limit the Number of threads.
- ❖ Where to put the cookie manager.
- ❖ Where to put the authentication manager
- ❖ Reducing the resource requirement
- ❖ Distributed testing.

9. Best Practices

Limit the Number of threads.

❖ This means the total number of threads generated by your test plan should be less than 300

❖ The overall number of threads running from BlazeMeter will be the total number of threads multiplied by the number of JMeter Engines.

❖ For example, a test plan with 200 threads and 4 JMeter Engines, will generate a load of 800 threads.

9. Best Practices

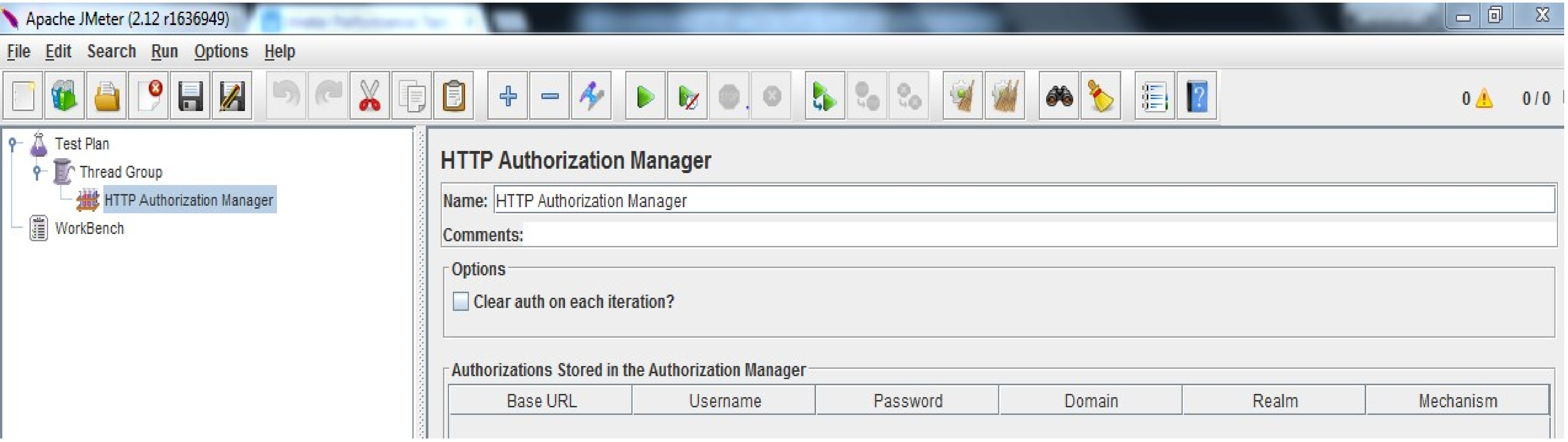
Where to put the cookie manager.

- ❖ The cookie manager store and sends cookies just like a web browser.
- ❖ Received cookies can be stored as JMeter thread variables.
- ❖ Manually add a cookie to the cookie manager.

9. Best Practices

Where to put the authentication manager

- It's species one or more user login for web page that are restricted using server authentication.


Apache JMeter (2.12 r1636949)

HTTP Authorization Manager

Name: HTTP Authorization Manager

Comments:

Options

Clear auth on each iteration? ☐

Authorizations Stored in the Authorization Manager

Base URL	Username	Password	Domain	Realm	Mechanism

9. Best Practices

Reducing the resource requirement

- ❖ Don’t use GUI Mode butter to use non- GUI mode.
- ❖ Use remote and distributed testing for large load testing.
- ❖ Do not load more than 300 threads per Jmeter.
- ❖ Use naming convention for all the elements.
- ❖ Use as few listeners as possible.
- ❖ Don’t use “view results tree” or “view results in table” listeners during the load test; use them only in scripting phase to debug purpose script
- ❖ Use CSV output rather than xml.
- ❖ Only save the data you need.
- ❖ Use few assertions as possible.