**A picture containing drawing

Description automatically generatedLearn from the Scratch: Step by Step**

**A picture containing bird

Description automatically generatedMd. Ataure Rahaman**

**SQA Engineer**

Contents

[Prerequisite: 3](#_Toc41075618)

[Basic Things: 3](#_Toc41075619)

[Tools available for recording JMeter UI test: 4](#_Toc41075620)

[Database test plan: 4](#_Toc41075621)

[Execute non-GUI mode: 4](#_Toc41075622)

[Html Report from Command Line: 5](#_Toc41075623)

[FTP- File transfer protocol: 5](#_Toc41075624)

[API: Application programming interface: 6](#_Toc41075625)

[API Call (Details): 6](#_Toc41075626)

[Data read from CSV: 6](#_Toc41075627)

[Setup Realistic Performance: 7](#_Toc41075628)

[Correlation & Regular Expression Extractor: 7](#_Toc41075629)

[HTTP Test Script Recorder: 8](#_Toc41075630)

[File Upload: 8](#_Toc41075631)

[File Download: 9](#_Toc41075632)

[Simple Login Record: 10](#_Toc41075633)

[JMeter - Selenium WebDriver integration: 10](#_Toc41075634)

[Remote Testing | Master Slave | Distributed Testing 11](#_Toc41075635)

[Server Health Monitor: 11](#_Toc41075636)

# Prerequisite:

1. Check java is installed in your machine (Java -version 7 or more)
2. Download jmeter form internet
3. Unzip it on your computer at any location
4. For starting jmeter click on (Jmeter/bin/jmeter.bat) file

# Basic Things:

1. **Thread Group**: That group are the users that will be used to create or run the test
   1. Sampler (To add request)
   2. Listener (to Receive the result)
2. **Assertion**: Check on the response
   1. Response Assertion
   2. Duration Assertion
   3. Size assertion
   4. HTML assertion
   5. XML assertion
   6. XPath assertion
3. **Listener**: Listener are the element in the JMeter that gather information about the test which are being executed.

latency = time to first byte

* 1. View Results in Table
  2. View Results Tree (Memory consuming)
  3. Aggregate Report
  4. Graph Results (Memory consuming)
  5. Summary Report
  6. Simple Data Writer

# Tools available for recording JMeter UI test:

* Badboy software (Windows)
* Blazemeter - Chrome Plugin - (Windows and Mac)

**Steps:**

1. Record a Test
2. Export as Jmeter (.jmx) Script
3. Open the script in Jmeter
4. Add listeners
5. Run and validate

# Database test plan:

* Add MySQL JDBC jar to JMeter lib folder -<https://dev.mysql.com/downloads/file/?id=492426>
* Add thread group
* Add JDBC connection config | provide details of your DB
* Add JDBC request
* Add listener
* Run and Validate

# Execute non-GUI mode:

* consumes more resources / memory
* not recommended for heavy load testing
* can be integrated with other systems -Jenkins/CI …

**Step 1:** Go to command line – go to JMeter - bin

**Step 2:** Command: JMeter -n -t (location of your JMeter test script) -l (location of the result file)

**-n:** non-GUI mode

**-t:** location of JMeter script

**-l:** location of result file

JMeter -h - to get help on JMeter commands

JMeter -? - to get information on JMeter command options

Html Report from Command Line: (This feature will work in JMeter 3.0 and later version)

Step 1: Create Test Plan and save it (and close).

Step 2: Open command line and change dir. to JMeter/bin

Step 3: Execute command:

**To create report at the end of the test:**

JMeter -n -t “location of the JMeter script” -l “location of result file” -e -o “location of the output folder”

**Create report from a standalone csv file:**

JMeter -g “location of csv file” -o “location of output folder”

Step 4: Analyze HTML (Dashboard) Reports

# FTP- File transfer protocol:

Step 1: Add an FTP Request Sampler

Step 2: Add FTP connection parameters

https://www.swfwmd.state.fl.us/data/ftp/

Server: [ftp.swfwmd.state.fl.us](ftp://ftp.swfwmd.state.fl.us)

Username: Anonymous (use your email address as password)

Step 3: Test an FTP GET and validate (get file from ftp server to local system)

Step 4: Test an FTP PUT and validate (transfer file from local to ftp server)

(FileZilla client is used in this demo for physical validation of file transfer. You can use other FTP clients like WinSCP)

**(To upload files with different names to FTP)**

Step 1: add CSV Data Set Config

Step 2: create a csv file and provide its location in CSV Data Set Config

Step 3: parametrize FTP PUT request to take value from csv file

Step 4: Run and Validate

# API: Application programming interface:

**Restaurant - table — WAITER — kitchen**

real world example - makemytrip.com

**Webservices - client — API — server**

REST | SOAP

How to test REST API

Step 1: Add HTTP Request Sampler OR Add SOAP/XML-RPC Request Sampler

Step 2: Add REST API details

Step 3: Run and Validate

API Call (Details):

<http://api.openweathermap.org/data/2.5/weather?q=Dhaka,%20BD&appid=884cb1eca4f082d98b840b41466538bc>

**IP**: api.openweathermap.org

**Path**: /data/2.5/weather/

**City**: q=Dhaka

**Parameter**: appid= 884cb1eca4f082d98b840b41466538bc

Account: <https://openweathermap.org/>

Credential: [ataur.214@gmail.com/api123456](mailto:ataur.214@gmail.com/api123456)

Keys: 884cb1eca4f082d98b840b41466538bc

# Data read from CSV:

Step 1: Add config element - CSV Data Set Config

Step 2: Add details in CSV Data Set Config

Step 3: Update value fields: ${variable\_name}

Step 4: Run and validate

# Setup Realistic Performance:

This plugin is deprecated and should not be used. Use Concurrency Thread Group instead. https://www.youtube.com/watch?v=\_YsZn1VwZYY

**What is a real-world performance test** Think Time - simulate actual user actions with timings/delays

**Pacing:**

* controlled ramp-up and down of virtual users
* control timing between iterations
* achieve x iterations in y mins/sec

Step 1 - Add Plugin - Stepping Thread Group

Step 2 - Setup load with required settings

Step 3 - Run and validate

**Timers:**

Purpose - to pause thread (v.user) for some time

* to add delay between threads
* to avoid over flooding the server and achieve real time

behaviour by pacing the load (to simulate v. user’s THINK TIME)

* Constant Timer
* Uniform Random Timer
* formula: 0.X \* Random Delay Max + Constant Delay Offset X : 0-9 example: 0.X \* 100 + 0 0 - 99 milli sec

# Correlation & Regular Expression Extractor:

Step 1: Create a Test Plan where you want to do dynamic referencing in JMeter

Step 2: Add Regular Expression Extractor in the Step from where response value(s) needs to be extracted

Step 3: Refer the extracted value (referred by Reference Name) into a subsequent step

Step 4: Run and validate references: <http://regexr.com/>

# HTTP Test Script Recorder:

**What is Test Script Recorder:** is a workbench element used to record user actions on browser

**How to record with Test Script Recorder:**

Step 1: Add Test Script Recorder - Non-Test Elements - HTTP(S) Test Script Recorder

Step 2: In a Thread Group add Logic Controller - Recording Controller

Step 3: Add the values in Test Script Recorder parameters

Step 4: Set Browser Proxy Configuration

Step 5: Install the certificate in your browser (if required)

Step 6: Start Recording

Step 7: Run and validate

**Helpful Tips**: Use JMeter’s inbuilt Recording template You can use the in-built template – Recording

# File Upload:

1. How to create test for file upload

2. How to record test for file upload

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**1 How to create test for File Upload**

Step 1: Create a Test Plan - Thread Group - HTTP Request

Step 2: Add values in HTTP Request sampler

Step 3: Add File Upload details

Step 4: Add Listeners to view results

Step 5: Run and Validate

**2 How to record test for File Upload**

Step 1: Add Template - Recording

Step 2: Provide a port number (e.g. 8181)

Step 3: Set browser to listen to this port

Step 4: Start Recording

Step 5: Filter the samples you need from recorded samples

Step 6: Run and Validate

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Helpful Tips:** In case you face any issues. Check the following : Method is POST Use multipart/form-data for POST Box is checked File location | Parameter | Mime Type are correct Check the logs for any other error In case you face any issues in recording the scenario... Watch : https://youtu.be/amEHuq8auTU?list=PLhW3qG5bs-L-zox1h3eIL7CZh5zJmci4c \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**References:** http://www.tinyupload.com/ <http://www.freeformatter.com/mime-types-list.html>

# File Download:

How to create test for file download

Step 1: Add Thread Group - HTTP Request sampler

Step 2: Add values in HTTP Request sampler

Step 3: Add Listeners to view results

Step 4: Add Listener - Save Responses to a file

Step 5: Add values to this Listener

Step 6: Run and Validate

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Helpful Tips How to avoid overwriting downloaded files. in a multi-user test Use

function: ${\_\_threadNum} as Prefix

References: <ftp://speedtest.tele2.net/>

# Simple Login Record:

Step 1: add BlazeMeter plugin to chrome browser

Step 2: start BlazeMeter plugin and login to BlazeMeter

Step 3: Record your scenario - Stop Recording - Export .jmx

In case you find jmx option disabled export as json and use this link to convert to jmx <http://converter.blazemeter.com/>

Step 4: Import imx file in JMeter

Step 5: Add listeners

Step 6: Run and validate

# JMeter - Selenium WebDriver integration:

Client-side performance analysis using WebDriver sampler

Step 1: Open JMeter

Step 2 : Add plugin JMeter > Plugin Manager > Selenium/WebDriver Support https://github.com/under6/jmeter-plugins-webdriver <https://github.com/under6/jmeter-plugins-webdriver/blob/m6ster/WebDriverTutori6l.wiki>

Step 3: Create a test plan > add thread group

Step 4: Add Config Element -> jp@gc - Chrome Driver Config S8mpler -> jp@gc - Web Driver S8mpler Listener -> View Results Tree

Step 5: Downlo6d chromedriver.exe 6nd provide the loc6tion in Chrome Driver Config e.g. - D:\Desktop\drivers\chromedriver\chromedriver.exe

Step 6: Add scripts in Web Driver Sampler

Step 7: Run & V6lid6te

Notes: WebDriver S6mpler 6utom6tes the execution 6nd collection of Perform6nce metrics on the Browser (clientside) While using WebDriver s6mpler e6ch thre6d will h6ve 6 single browser inst6nce 6nd e6ch browser consumes signific6nt 6mount of resources

# Remote Testing | Master Slave | Distributed Testing

Step 1: Setup Master

* Added remote systems IP in jmeter.properties

Step 2: create KeyStore file

* run create-rmi-keystore.bat
* name: rmi
* password: changeit

Step 3: run jmeter-server file on sl.ve (remote) system

Step 4: Run and Validate

Helpful Tips:

* All systems (master and slaves) have same ver of JMeter
* All systems have java (preferably same ver)
* All systems can connect to each other (are in same subnet)
* No need to copy jmeter script (jmx) to slave systems
* If you want to have 100 users and using 2 slaves. Give no as 50

# Server Health Monitor:

Server - machine where your test application is hosted CPU, Memory, Disk I/O, Network

Step 1: Add perfmon plugin in JMeter

* Download and add to JMeter lib and ext folder <https://jmeter-plugins.org/wiki/PerfMon/>
* Plugins Manager

Step 2: On server > Download perform server agent

Step 3: Start the perfmon server agent

* Windows - startAgent.bat
* Check client can talk to server via port 4444

Step 4: Create JMeter test to monitor Server Health

Step 5: Run and Validate