STANDARD OPERATING PROCEDURE

Setup Guide for Performance Testing

1. **Purpose:** The purpose of this SOP is to provide detailed guidelines for installing the Performance Testing tool. This document also includes step-by-step instructions for conducting performance testing, covering setup configurations, execution, and analysis. These steps ensure accurate testing and reliable results, helping users effectively assess and optimize system performance.

2. **Scope:** This SOP outlines all essential steps, tools, and processes for setting up and conducting Performance Testing, from installation and configuration to executing tests and analysing results for optimal system performance.

Document Details	
Reference Number	SOP-PT-001
Version	0.1
Effective Date	01-NOV-2024
Prepared By	Vinayak
Review By	Ankur Kumar
Approved By	

Contents

Objective

- The objective of performance testing is to evaluate and ensure a system's responsiveness, stability, scalability, and resource usage under specific workloads.
- It helps identify bottlenecks, establish baseline metrics, and verify that applications meet performance expectations before they go live.
- Key goals Include: - Validating Speed, Ensuring Stability, Evaluating scalability, Resource Usage Optimization.

Performance Testing Tool

JMeter is a **pure Java** application and should run correctly on any system that has a compatible Java implementation.

Here is the list of an operating system compatible with JMeter: -

- Linux
- Windows
- Mac OS
- Ubuntu

Steps to Install JMeter

Step 1) Install Java

Because JMeter is pure Java desktop application, it requires a fully compliant JVM 6 or higher.

You can download and install the latest version of Java SE Development Kit. Download Java Platform (JDK)

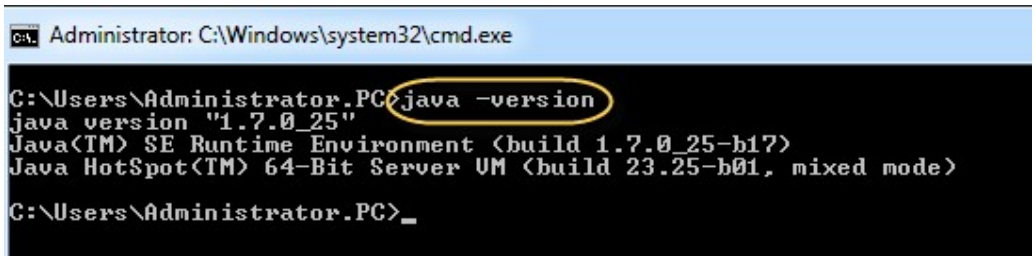
After installation is finished, you can use the following procedure to check whether Java JDK is installed successfully in your system.

- In Window/Linux, go to **Terminal**
- Enter command **java -version**

If the Java runtime environment is installed successfully, you will see the output as the figure below:-

If nothing displays, please re-install Java SE runtime environment.

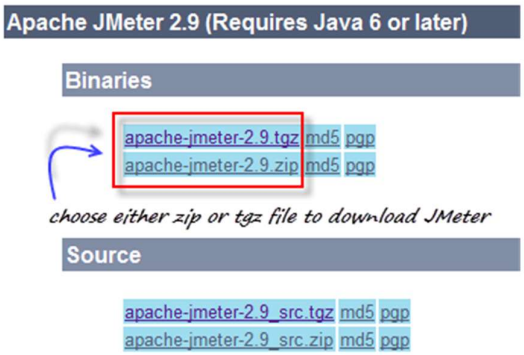
Please see the link for details instructions https://www.guru99.com/install-java.html



Step 2) Download Jmeter

As of this writing, the latest version of JMeter is **Apache JMeter 4.2**. You can download it here But this tutorial demos installation of version 2.9, the install process remains the same.

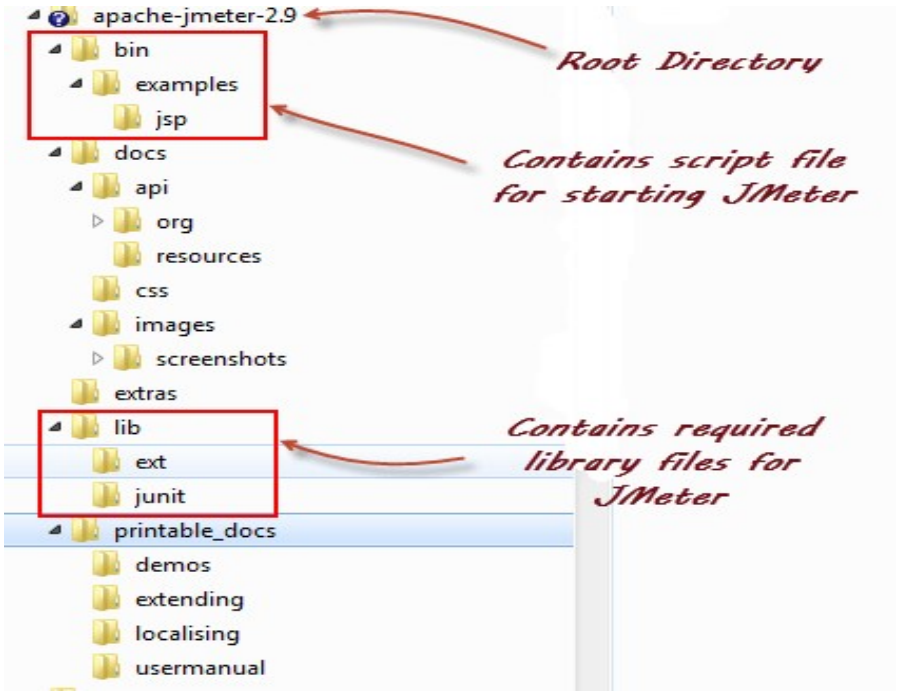
Choose the Binaries file (either zip or tgz) to download as shown in the figure below.



Step 3) Installation

Installation of JMeter is extremely easy and simple. You simply unzip the zip/tar file into the directory where you want JMeter to be installed. There is no tedious installation screen to deal with! Simply unzip and you are done!

Once the unzipping is done installation directory structure should look like as figure below:



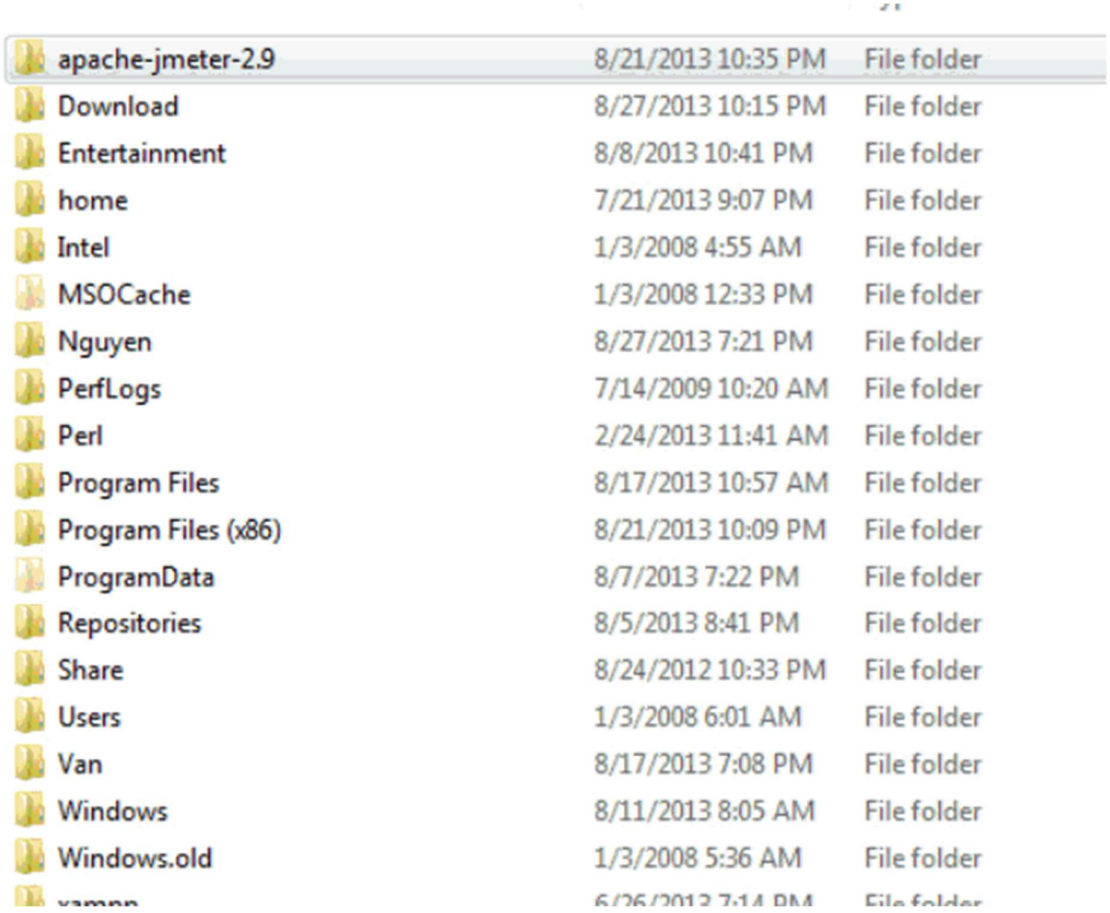
Launch JMeter

You can start JMeter in 2 modes: -

- GUI Mode
- Command Line Mode

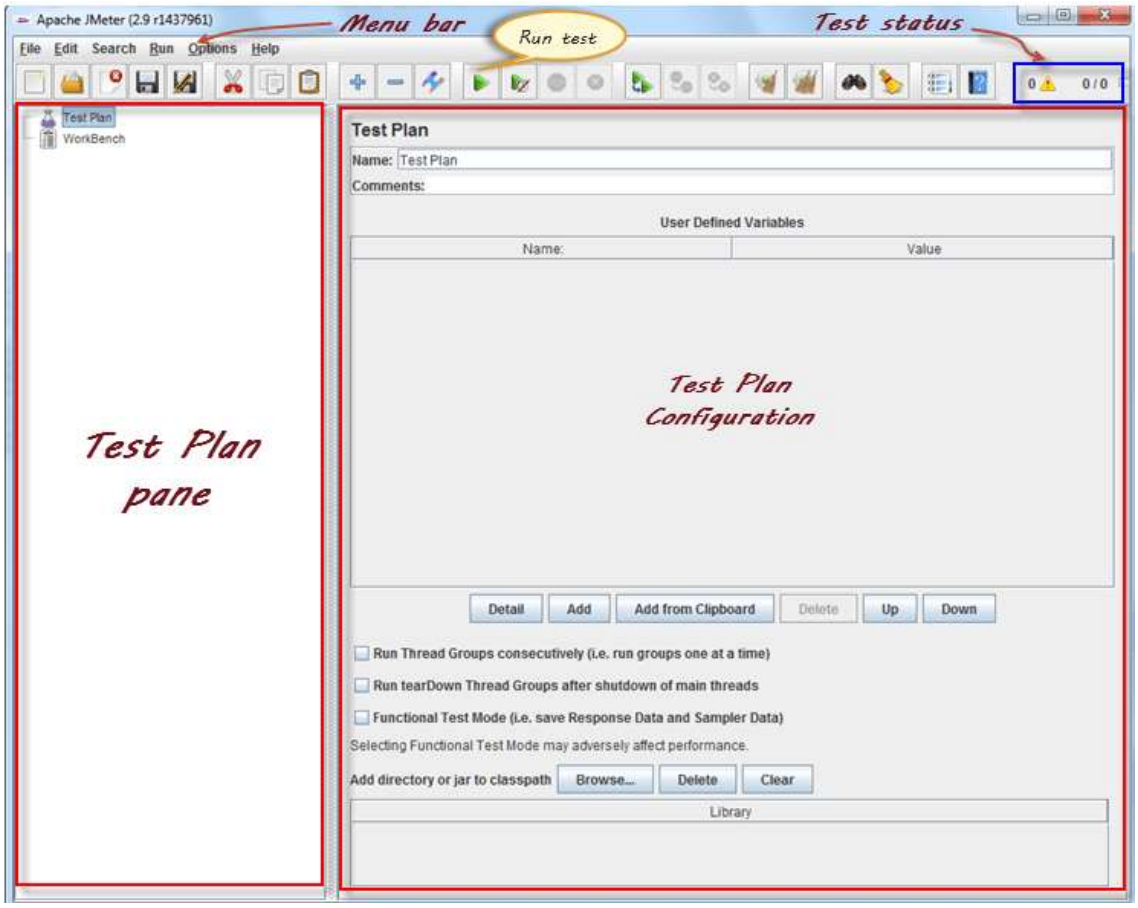
Start JMeter in GUI Mode

- If you are using Window, just run the file **/bin/jmeter.bat** to start JMeter in GUI mode as shown below.



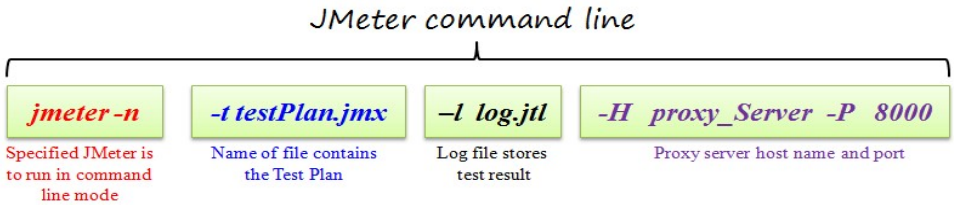
The

Following figure annotates the various components in the JMeter GUI



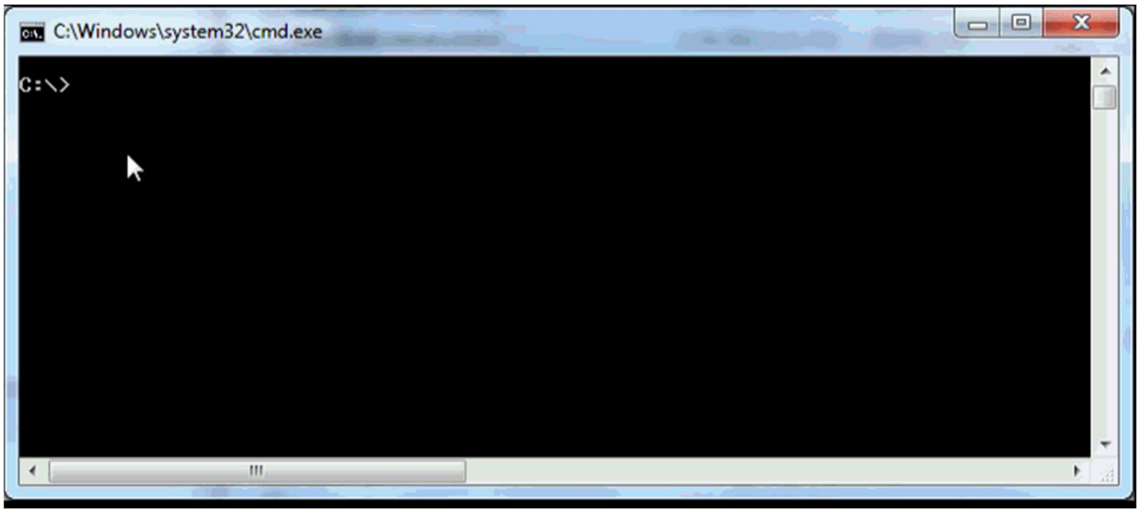
Start JMeter in command line mode

- JMeter in GUI mode consumes much computer memory. For saving the resource, you may choose to run JMeter without the GUI. To do so, use the following command options.



This is a command line example

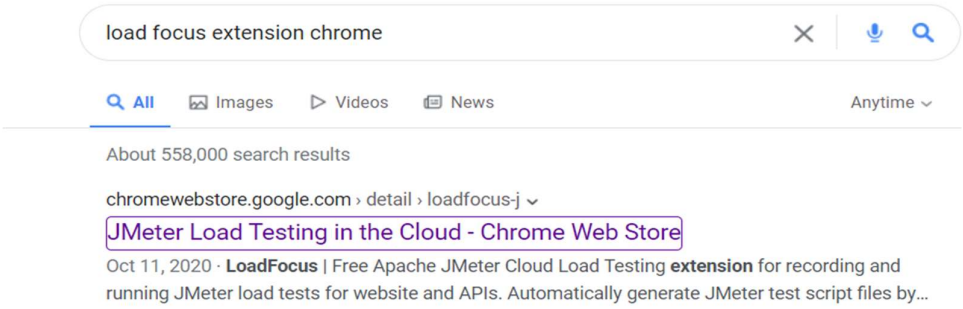
$jmeter -n -t testPlan.jmx - l log.jtl -H 127.0.0.1 -P 8000



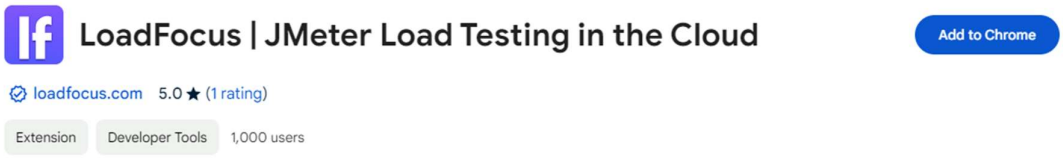
Setting Up Load Focus JMeter Chrome Extension

After successfully installing JMeter, you can enhance your testing setup by integrating the **Load Focus JMeter Chrome Extension**. This extension allows you to simulate and record user actions directly within your browser, making it easier to capture performance metrics.

Step 1: - Hit the first link.



Step2: - By clicking Add to chrome, Extension will be added.



Testing Procedure

1) Launch the Application: Open the web application you wish to test. Ensure that the application is accessible and functioning correctly before initiating any performance tests.

2) Recording Scenario 1 –

User Login:

a. Begin by starting a new session to test the login functionality.

b. Enter the necessary login credentials:

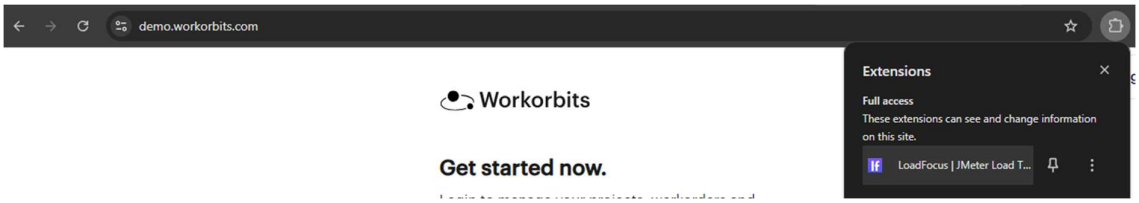
- Username: Input the appropriate username for the login test.
- Password: Enter the corresponding password.

3) Start Recording with Load Focus:

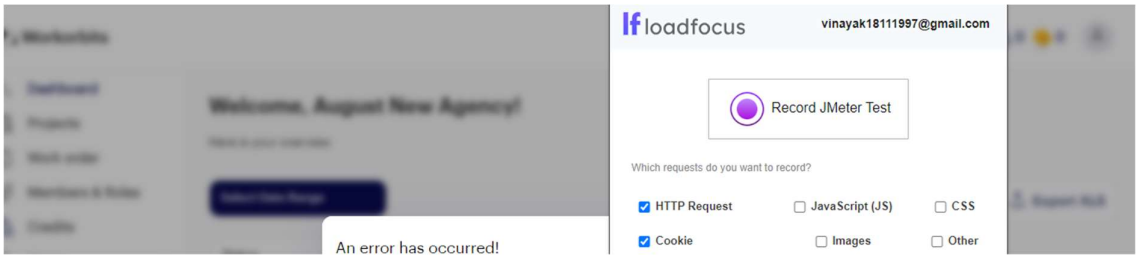
- Before clicking the "Login" button, activate the Load Focus extension to begin recording the interaction.
- The extension will track the response time, capturing the duration it takes for the application to process the login request.

How to activate the Load focus extension: -

1) Click on the Icon shown in the Image below.



2) Press Record JMeter test button.

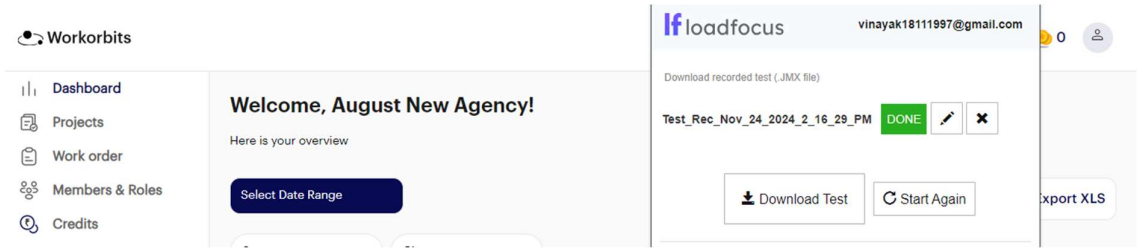


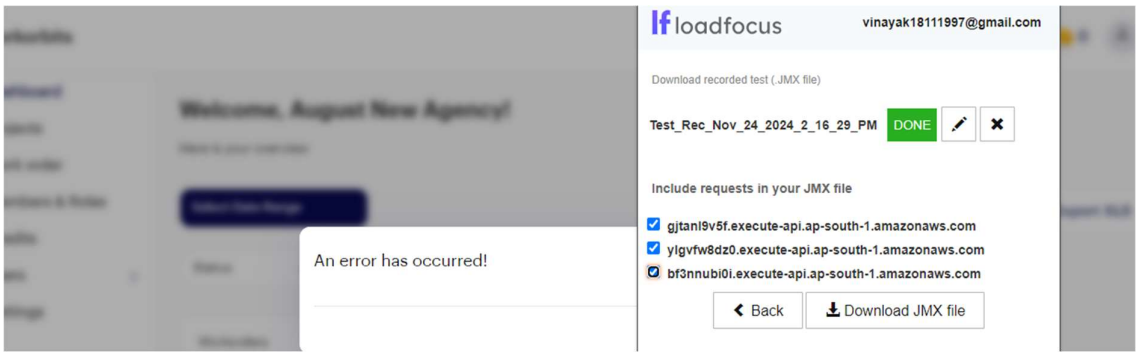
3) Complete the Login Action:

- Click on the "Login" button while the Load Focus recording is active.
- Observe and record the response time metrics captured by the Load Focus tool. This data will provide insights into how quickly the application handles login requests under normal conditions.

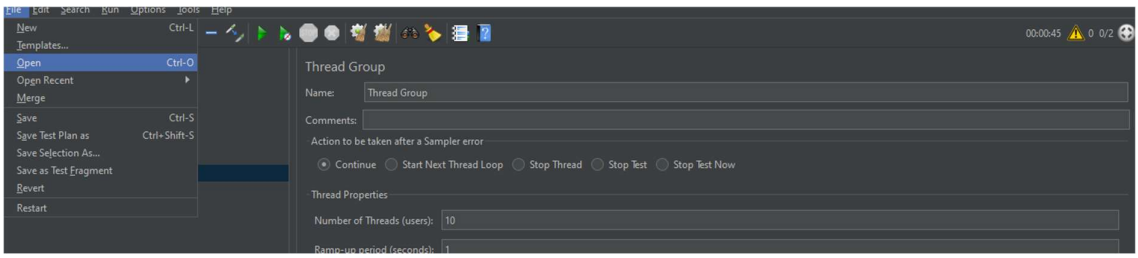
4) **Download and Clean the JMX File:**

- Once the recording is complete, download the JMX file from Load Focus. The JMX file contains the recorded API calls and user interactions in a format compatible with JMeter.
- After clicking on the Download Test, Tick marks the boxes as shown in the image.
- Then click on the Download the JMX file button to download it.





- Open the downloaded JMX file in JMeter and review the list of recorded APIs.

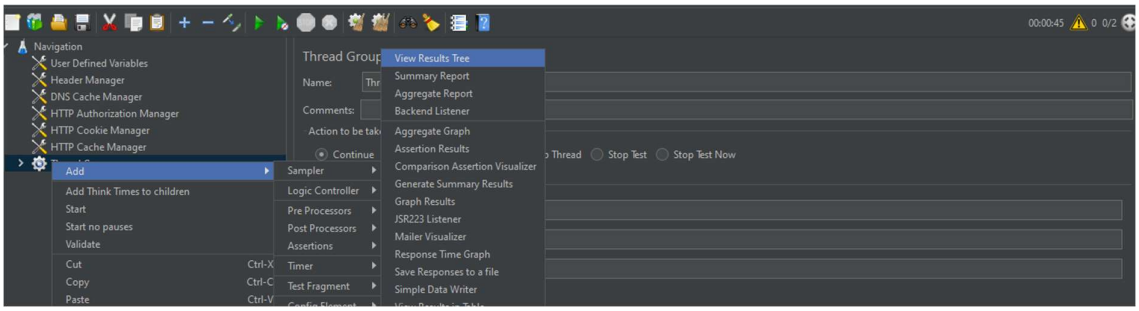


- Remove any unnecessary or extra APIs that are irrelevant to the login scenario.
  This step ensures that the test script remains focused on the primary actions being tested.

- The "View Results Tree" listener in JMeter is a vital tool used during performance testing

Key benefits of View Result Tree: -

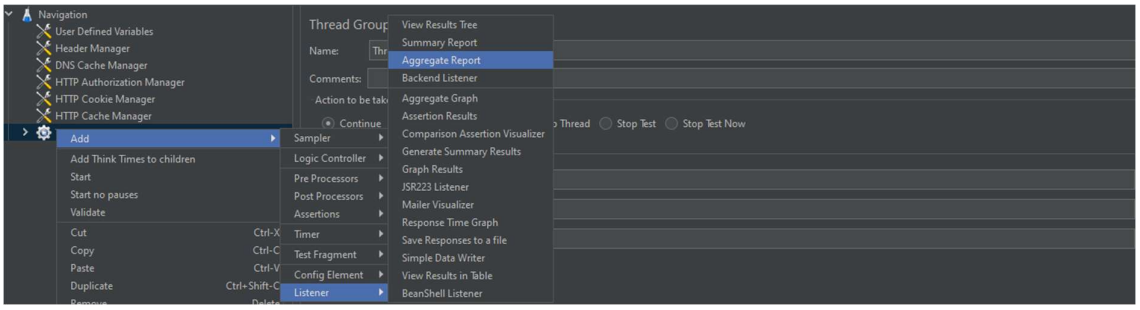
1) Debugging Test Scripts
2) Validating the Correctness of Test Cases
3) Monitoring Response Times and Latency
4) Viewing Errors and Failures



- The Aggregate Report in JMeter is a powerful listener that provides a summarized view of the performance test results. It is widely used for analysing and interpreting the test data, especially when dealing with large-scale performance testing scenarios.

Key benefits of View Result Tree: -

1) Summarized Metrics Overview
2) Performance Analysis and Comparison
3) Error Rate and Stability
4) Exportable Data for Further Analysis



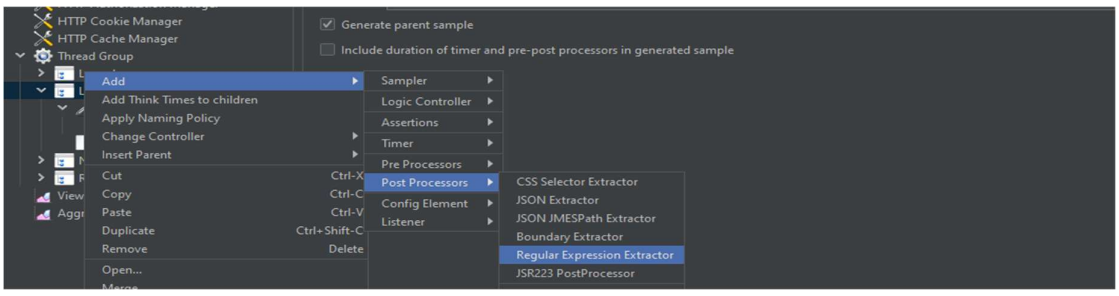
Tokenization – Handling Dynamic Data

In JMeter, handling tokenization with dynamic data typically involves extracting tokens from server responses and reusing them in subsequent requests. Here's a step-by-step approach to achieve this:

1. Add a Regular Expression Extractor:

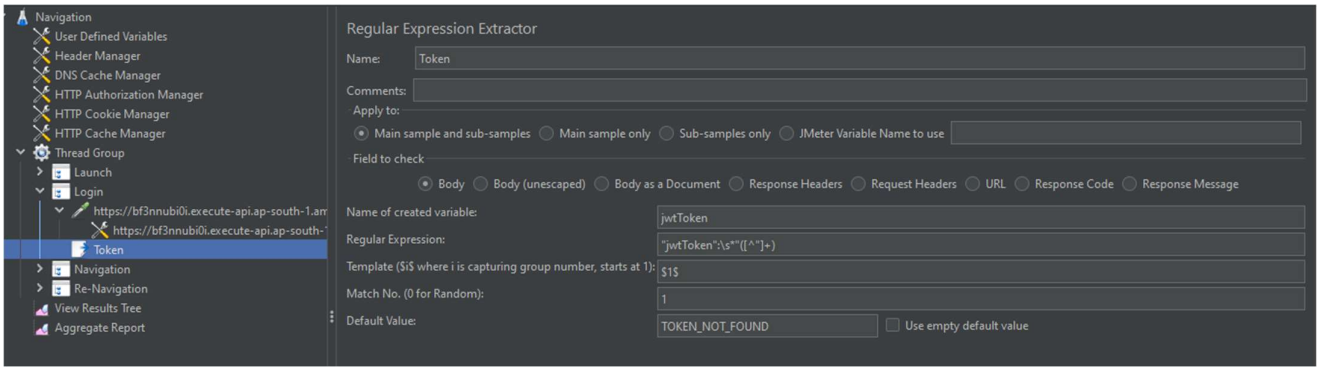
Purpose: To extract dynamic tokens like JWT or CSRF tokens from server responses.

- **Steps**:
    - Add a **Regular Expression Extractor** to the HTTP Request where the token is expected in the response.
    - Set the **Response Field to Check** to Body, Headers, or whichever field contains the token.
    - In **Regular Expression**, use your pattern like:
    - Set a **Reference Name** (e.g., jwtToken), which will hold the extracted token.
    - Set the **Template** as 1 to capture the first group in the regular expression.



2. Reuse the Extracted Token:

- Now, you need to use the extracted token in subsequent HTTP requests. You can reference it using ${jwtToken} wherever you want in the request headers or parameters.
- For example, if you need to pass it as an Authorization header in the next request:
    - Add an **HTTP Header Manager** to the next HTTP Request.
    - Add a header with **Name** as Authorization and **Value** as Bearer ${jwtToken}.



3. Configure the JMeter Test Plan:

- Make sure the **Regular Expression Extractor** is correctly placed under the HTTP request that receives the token.
- Ensure all subsequent requests that need the token have it included in their header or body using ${jwtToken}.

4. Parameterize Dynamic Data:

If you have dynamic data that changes frequently (like user credentials, URLs, or request bodies):

- Use **CSV Data Set Config** to read external data files with dynamic parameters.
- Reference these parameters similarly using ${variableName} wherever needed in your HTTP requests.

Example Test Plan Flow:

1. **Login Request**: Extract a JWT token from the server response.
    - Add **HTTP Request** for login.
    - Add **Regular Expression Extractor** to extract "jwtToken".
2. **Subsequent API Requests**: Use the extracted token in the headers.
    - Add **HTTP Header Manager** to each HTTP request requiring authorization.
    - Set **Authorization** as Bearer ${jwtToken}.

Key Considerations:

- Test your Regular Expression separately to ensure it captures the token correctly.
- Debug Sampler can be used to view the extracted variables during test execution.
- If tokens expire after a certain period, ensure your test plan requests a new token when necessary.