Understanding Performance Testing: A Beginner's Guide

Chapter 4

**JMeter: A Comprehensive Performance Testing Tool**

**Test Plan**

* A Test Plan is essentially the root of your JMeter script.
* It's a container for all the elements required to execute your performance test. Think of it as the blueprint for your entire test scenario.
* It includes all the elements required for a test, such as thread groups, logic controllers, samplers, listeners, timers, assertions, and configuration elements.

**Key components of a Test Plan**

**1 Thread Groups**

* Define the number of users (threads) and the ramp-up period (time to create all threads) for your test. Each thread simulates a real user.
* You can also set the number of iterations each thread will perform.

**2 Samplers**

* Responsible for sending requests to the server. Common samplers include HTTP Request, FTP Request, JDBC Request, and JMS Request.

**3 Logic Controllers**

* Determine the order of execution of Samplers. Examples include Loop Controller, If Controller, and While Controller.

**4 Listeners**

* Gather and display test results. Examples include View Results Tree, Graph Results, and Summary Report.

**5 Timers**

* Introduce delays between requests to simulate real user interactions more accurately.

**6 Assertions**

* Verify that the server response meets the expected criteria. Common assertions include Response Assertion, Duration Assertion, and Size Assertion.

**7 Configuration Elements**

* Set up defaults and variables for your test. Examples include HTTP Request Defaults, CSV Data Set Config, and User Defined Variables.

**Workbench**

* The Workbench is a temporary storage area for test elements.
* It's a place to experiment with different components without affecting the Test Plan. Any elements added to the Workbench are not saved when you save the Test Plan.
* The Workbench in JMeter is a non-executable portion of the test plan.

**Key purpose of Workbench**

**Temporary Storage**:

* It is used to store test elements temporarily. Elements in the Workbench are not saved with the test plan by default and are not executed during the test run.

**Testing Elements**:

* You can test out various configurations and elements without affecting the main test plan.

**GUI Elements**:

* Useful for elements that do not have a direct impact on the execution of the test plan but are needed for building or debugging the test.

**Saving Workbench**:

* By default, the Workbench is not saved with the test plan.

**Experimentation:**

* Test different components before adding them to the Test Plan.

**Note:** In latest version, Workbench option is removed and to access its functionality. You can directly right click on Test Plan -> NonTestElements -> HTTP(S)TestScriptRecorder

**Record and Playback in JMeter**

Record and Playback is a feature in JMeter that allows you to capture user interactions with a web application and generate a basic test script automatically. This can be a quick way to get started with building performance tests. This is particularly useful for those who are not familiar with writing test scripts manually. It allows users to record their interactions with a web application and then replay these interactions during testing.

**Recording in JMeter**

1. **Creating a Test Plan:**
   * Start by creating a new Test Plan.
2. **Configuring the HTTP(S) Test Script Recorder:**
   * Add an HTTP(S) Test Script Recorder to the Test Plan. This component acts as the proxy server and captures the traffic.
   * Configure the Test Script Recorder by setting the port number and other settings like URL patterns to include or exclude certain requests.

**Note:** To add the HTTP(S) Test Script Recorder, Open the JMeter Tool and by default you will have the Test Plan. Click on Test Plan -> Add -> Non-Test Elements -> HTTP(S) Test Script Recorder.

1. **Setting Up the Proxy Server:**
   * JMeter can act as a proxy server to capture HTTP or HTTPS requests. To do this, you need to configure your browser to use JMeter as a proxy.

**Note:**

1. On HTTP(S) Test Script Recorder, Under Global Settings section, you can see the **Port** with default value **8888**. You can use it or change it as per your need.
2. Under Test Plan Creation section, choose the **Target Controller** as **Test Plan > HTTP(S) Test Script Recorder** .
3. Under Requests Filtering section, select **Add suggested Excludes** from **URL Patterns to Exclude**.
4. Click on Start button and stop it. Now, the JMeter HTTP(S) Test Script Recorder has generated the Certificate under bin folder.

Example: **ApacheJMeterTemporaryRootCA**

**5. Steps to configure Proxy on Firefox**

* + Open the Firefox browser
  + Click on Hamburger icon and go to Settings
  + Search for Proxy in search bar
  + Click on Network Settings
  + Connection Settings Dialog should open
  + Choose Manual Proxy configuration
  + Enter localhost on HTTP Proxy and 8888 on Port
  + Check the “Also use this proxy for HTTPS” checkbox
  + Click Ok

**6. Steps to configure Certificates on Firefox**

* + Open the Firefox browser
  + Click on Hamburger icon and go to Settings
  + Search for Certificates in search bar
  + Click on View Certificates
  + Certificate Manager Dialog should open
  + Click on Import button and select the created Certificate from JMeter HTTP(S) Test Script Recorder
  + Click OK

1. **Recording the Test:**
   * Start the HTTP(S) Test Script Recorder.
   * Launch the browser on navigate to the test website, perform the actions you want to test in your web application using the browser configured to use JMeter as a proxy.
   * The actions are recorded as HTTP requests under the Target Controller (HTTP(S) Test Script Recorder).

**Playback in JMeter**

1. **Review and Modify the Recorded Script:**

* After recording, review the recorded HTTP requests. You can modify them as needed, such as parameterizing values, adding assertions, or inserting timers.

1. **Adding Listeners:**

* Add listeners to the Test Plan to capture and display the results of the test execution. Common listeners include View Results Tree, Summary Report, and Aggregate Report.

1. **Running the Test:**

* Once the test plan is set up and modified as needed, you can execute the test.
* JMeter will replay the recorded interactions with the web application, simulating multiple users if configured in the Thread Group.

**Advantages of Record and Playback**

1. **Ease of Use:**

* Simplifies the process of creating test scripts, especially for beginners or those who prefer not to write scripts manually.

1. **Time-Saving:**

* Quickly creates a baseline test script that can be modified and enhanced as needed.

1. **Accuracy:**

* Captures exact user interactions with the application, reducing the chances of missing critical requests or parameters.

**Limitations**

**Initial Setup:**

* Requires initial setup of the proxy server and browser configuration, which might be complex for some users.

**Maintenance:**

* Recorded scripts may require significant maintenance, especially if the application under test changes frequently.

**Performance Overhead:**

* Recording can introduce some performance overhead, especially if a large number of requests are being captured.

**Basic Script:**

* The recorded script is often a starting point and may require manual adjustments to accurately simulate user behavior.

**Dynamic Content:**

* JMeter might struggle with handling dynamic content, requiring manual correlation.

**Complex Scenarios:**

* Complex user interactions might not be captured accurately.

**When to Use:**

* Quick Script Creation: For simple test cases or to get a basic understanding of the application flow.
* Learning JMeter: To familiarize yourself with JMeter components.