APACHE
JMeter™

TESTING TOOL

Author:

LAKSHMAN PRABHU.B
23MZ37

INTRODUCTION

- Open-source nature under the Apache License 2.0.
- Java-based architecture, platform-independent.
- Primary use is to assess the performance and load handling of applications.
- Comprehensive features for load testing, stress testing, and performance testing.
- Supports a wide range of protocols, including HTTP, HTTPS, FTP, JDBC, SOAP, etc.

WHY JMETER?

- Intuitive interface suitable for both beginners and experienced testers.
- Well-documented with a strong user community for learning and troubleshooting.
- Applicable to load testing, stress testing, and performance monitoring.
- Integrates into CI/CD pipelines for automated performance testing.
- Supports distributed testing for scaling tests across multiple machines.

FEATURES OF JMETER?

- Thread Groups: Define the number of users and the way they should behave during the test.
- Samplers: Simulate different types of requests, such as HTTP requests, FTP requests, JDBC requests, etc.
- Listeners: Collect and display the results of the test in various formats, aiding in analysis.
- Controllers: Manage the flow of the test plan, allowing for conditional and loop-based execution.
- Timers: Introduce delays between requests, ensuring a more realistic simulation of user behavior.

PROCEDURE FOR BEGINNERS TO INSTALL JMETER

1. **Download JMeter:**
   - Visit the official Apache JMeter website (https://jmeter.apache.org/).
   - Look for the "Download" section and select the latest stable version.
   - Download the binary file (e.g., ApacheJMeter-5.4.1.zip for Windows).
2. **Extract the Downloaded File:**
   - Once the download is complete, extract the contents of the zip file to a location on your computer.
3. **Verify Java Installation:**
   - JMeter is a Java-based application, so ensure that Java is installed on your machine.
   - Open a command prompt (Windows) or terminal (Mac/Linux) and type java -version to check if Java is installed and the version.

PROCEDURE FOR BEGINNERS TO INSTALL JMETER

4. **Set Java Environment Variables (if needed):**
   - If Java is not in your system's PATH, set the JAVA_HOME environment variable to the directory where Java is installed.
5. **Run JMeter:**
   - Navigate to the bin directory within the extracted JMeter folder.
   - Run the jmeter.bat (Windows) or jmeter.sh (Mac/Linux) script to start JMeter.
6. **Verify Installation:**
   - JMeter's graphical user interface should open. You should see the main JMeter window.

PROCEDURE FOR BEGINNERS TO INSTALL JMETER

7. **Optional: Increase Heap Size (if needed):**
   - For large tests, you might need to increase JMeter's heap size.
   - Open the jmeter.bat (Windows) or jmeter.sh (Mac/Linux) file in a text editor and modify the -Xms and -Xmx values.
8. **Explore Documentation:**
   - Familiarize yourself with the official JMeter documentation available on the website. It provides in-depth information on using JMeter effectively.

Test Plan Creation - Building a Test Plan

- ❑ **Concept of Test Plan:**
  - ▪ Understand that a test plan is a blueprint for your performance test.
- ❑ **Adding Thread Groups:**
  - ▪ Open JMeter, right-click on the Test Plan, and add a Thread Group.
- ❑ **Configuring Samplers:**
  - ▪ Within the Thread Group, add a Sampler (e.g., HTTP Request) to simulate user actions.
- ❑ **Setting Up Listeners:**
  - ▪ Add a Listener to capture and view test results.
- ❑ **Basic Test Plan Creation:**
  - ▪ Arrange Thread Groups, Samplers, and Listeners to create a basic test plan.

Recording Tests in JMeter

- **Using JMeter's Proxy Server:**
  - Start the JMeter Proxy Server in JMeter.
- **Configuring Proxy Settings:**
  - Set up your browser's proxy settings to point to JMeter's Proxy (localhost:8080).
- **Recording User Interactions:**
  - Perform actions in the browser, and JMeter will record the interactions.
- **Stopping the Recording:**
  - Stop the recording in JMeter when the test scenario is complete.
- **Reviewing Recorded Requests:**
  - Examine the recorded requests in JMeter to ensure accuracy.

Parameterization in JMeter

- ❑ **Introduction to Parameterization:**
  - ▪ Understand parameterization as a way to create dynamic tests.
- ❑ **Using Variables:**
  - – Learn how to use variables to parameterize values in your test.
- ❑ **CSV Files for Data:**
  - ▪ Explore how to use CSV files to supply data for parameterization.
- ❑ **Other Parameterization Techniques:**
  - ▪ Discover additional techniques for dynamic test data.
- ❑ **Implementing Parameterization:**
  - ▪ Apply parameterization to make your tests more realistic.

Performance Monitoring - Monitoring Performance Metrics

- **JMeter's Monitoring Capabilities:**
  - Understand how JMeter can monitor performance metrics during tests.
- **Importance of Monitoring:**
  - Highlight the significance of monitoring to identify bottlenecks.
- **Configuring Monitors:**
  - Configure JMeter to collect and display performance metrics.
- **Real-time Monitoring:**
  - Monitor performance metrics in real-time during test execution.
- **Analyzing Bottlenecks:**
  - Use performance monitoring to identify and analyze bottlenecks.

Reporting and Analysis - Analyzing Test Results

- ❑ **JMeter's Reporting Features:**
  - ▪ Explore the reporting and analysis features in JMeter.
- ❑ **Introduction to Listeners:**
  - ▪ Introduce different Listeners for viewing and analyzing test results.
- ❑ **Interpreting Results:**
  - ▪ Provide examples of how to interpret and analyze test results.
- ❑ **Generating Reports:**
  - ▪ Generate comprehensive reports for in-depth analysis.
- ❑ **Improving Test Scenarios:**
  - ▪ Use analysis to refine and improve test scenarios.

Integrations and Plugin's - Integrating JMeter with Other Tools

- ❑ **CI/CD Tool Integration:**
  - Explore how JMeter can be seamlessly integrated into Continuous Integration/Continuous Deployment (CI/CD) pipelines for automated performance testing.
- ❑ **Version Control System Integration:**
  - Discuss the integration of JMeter with version control systems like Git, enabling collaborative test script development and version tracking.

Integrations and Plugins - Integrating JMeter with Other Tools

❑ CI/CD Tool Integration:

- Explore how JMeter can be seamlessly integrated into Continuous Integration/Continuous Deployment (CI/CD) pipelines for automated performance testing.

❑ Version Control System Integration:

- Discuss the integration of JMeter with version control systems like Git, enabling collaborative test script development and version tracking.

Applications of Jmeter Tool

- **E-commerce Websites** - to simulate user loads
- **Financial Services** - to test the performance
- **Healthcare Systems** - handle concurrent access
- **Media Streaming Services** - ability to handle varying levels of user engagement
- **Gaming Applications** - to guarantee a smooth gaming experience for users
- **API Testing** - performance and reliability of APIs in various domains

Thank You