



InterviewBit

JMeter Interview Questions



To view the live version of the page, [click here](#).

© Copyright by Interviewbit

Contents

JMeter Interview Questions for Freshers

1. Discuss some of the features of JMeter.
2. What is Distributed Testing?
3. In JMeter, how do you set up a master-slave configuration?
4. Explain the usage of Regular Expression in JMeter. Also discuss the difference between 'contains' and 'matches'.
5. Discuss about Samplers and Thread Groups in JMeter.
6. What are the different types of processors available in JMeter?
7. Mention the order of execution of JMeter test plan elements.
8. Explain configuration elements in JMeter.
9. What are the various data parameterization options in JMeter?
10. What is assertions in JMeter?
11. What is the maximum number of threads that should be allowed on a single system?
12. Discuss Gaussian and Poisson Timers.
13. In JMeter, explain the purpose of correlation.
14. What are the various kinds of listeners?
15. What exactly is a workbench?
16. Discuss the working of the Test Script Recorder.
17. What protocols are supported by JMeter?
18. Explain the JMeter variable and function syntax.

JMeter Interview Questions for Experienced

JMeter Interview Questions for Experienced

(.....Continued)

19. Is it possible to use JMeter to record actions using a mobile device? If so, how would you go about doing it?
20. Why is running JMeter in Non-GUI mode recommended?
21. How do you perform spike testing with JMeter?
22. Is it possible for you to use JMeter to run Selenium scripts? If so, how would you go about doing it?
23. In JMeter, how do you handle sessions and cookies?
24. What are the crucial steps in the JDBC (Java Database Connectivity) request testing process?
25. What is BeanShell scripting?
26. Write the code to write data stored in a JMeter variable to a CSV file.
27. What is a Root Certificate Authority (CA) certificate?
28. What factors influence the number of threads that should be generated per system?
29. What are the most significant plugins that JMeter supports?
30. In JMeter, what are the different sorts of controllers?
31. How is Ultimate thread group different from other thread groups?

Let's get Started

What is JMeter?

The Apache JMeter application is free and open-source software that is made entirely of [Java](#) and is used to load-test and measure the performance of applications and different software products. It was created to test Web applications, but it has now been expanded to include other test functions like performance testing, load testing and stress testing. JMeter can be used to test the performance of both static and dynamic assets, as well as Web dynamic applications. It can be used to simulate a heavy demand on a server, set of servers, network, or item in order to test its strength or examine the entire performance under various load scenarios.

JMeter is a protocol analyzer, not a browser. JMeter appears to be a browser (or rather, several browsers) when it comes to web services and remote services, but it does not do all of the operations that browsers do. JMeter, in particular, does not run Javascript present in HTML pages. It also doesn't produce HTML pages like a browser (you can examine the answer as HTML and so on, but the timings aren't included in any samples, and only one sample from each thread is ever displayed at a time).

[Learn More](#).



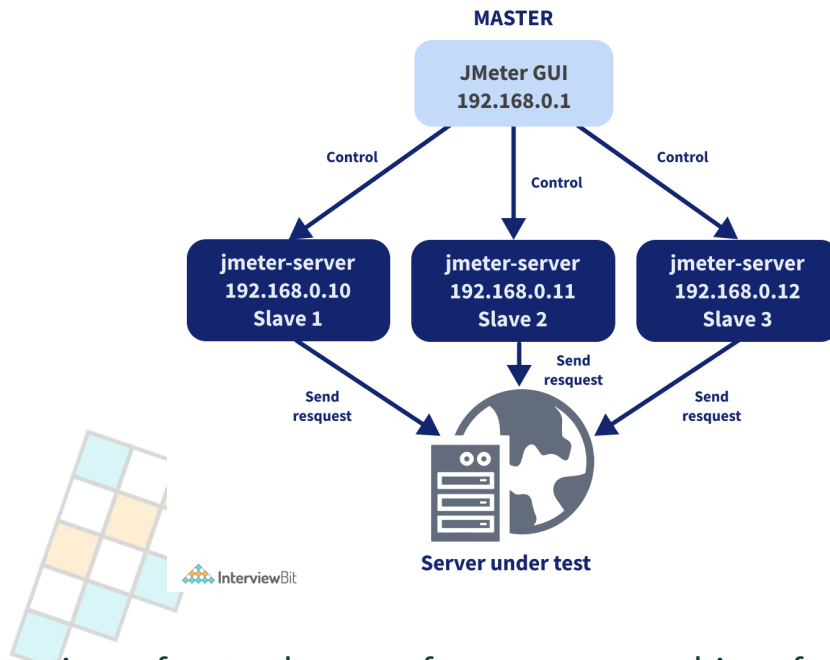
JMeter Interview Questions for Freshers

1. Discuss some of the features of JMeter.

The following are some of the features of Apache JMeter:

- The ability to load and performance test a wide range of apps, servers, and protocols. This includes the following :
 - Web - HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET, ...)
 - SOAP (Simple Object Access Protocol)/ REST (Representational State Transfer) Web Services
 - FTP (File Transfer Protocol)
 - Database via JDBC (Java Database Connectivity)
 - LDAP (Lightweight Directory Access Protocol)
 - Message-oriented middleware (MOM) via JMS (Java Message Service)
 - Mail - SMTP (Simple Mail Transfer Protocol)(S), POP3 (Post Office Protocol 3)(S) and IMAP (Internet Message Access Protocol)(S)
 - Native commands or shell scripts
 - TCP (Transmission Control Protocol)
 - Java Objects
- Full-featured Test IDE for recording (from browsers or native applications), developing, and debugging Test Plans (A Test Plan is where you add elements that are needed for your JMeter Test.)
- To load-test from any Java-compatible OS (Linux, Windows, Mac OSX,...), use CLI mode (Command-line mode (formerly termed Non-GUI) / headless mode).
- It has the option to generate a dynamic HTML report that is full and ready to present.
- The ability to extract data from the most popular response formats, such as HTML, JSON, XML, or any textual format, allows for easy correlation.
- Complete portability and Java purity are guaranteed in Jmeter.
- Concurrent sampling by several threads and simultaneous sampling of distinct functions by separate thread groups are both possible with a full multithreading framework.
- It also has the option of test results caching and analysis/replaying offline.
- Extensible core:
 - Pluggable Samplers offer limitless testing capabilities.
 - With pluggable timers, you can choose from a variety of load statistics.
 - Plugins for data analysis and visualisation offer a lot of flexibility and personalization.
 - The employment of functions can be utilised to offer dynamic input to a test or to manipulate data.
 - 3rd party Open Source libraries for Maven, Gradle, and Jenkins make Continuous Integration simple.

2. What is Distributed Testing?



Distributed testing refers to the use of numerous machines for load testing (Load testing is the technique of simulating several users accessing a software at the same time in order to mimic the expected usage of the application), with one serving as the master and the others as slaves. It is critical to notice that all machines must be connected to the same network and run the same version of Java and JMeter.

We'll utilise a local client as a master to manage the test execution for multiple remote clients, and each remote client will act as a slave to run the test on our target server. Each slave system runs the load tests according to the master's exact device specifications. As a result, distributed performance testing enables us to handle a greater number of concurrent users contacting the target server.

3. In JMeter, how do you set up a master-slave configuration?

A master-slave configuration is a type of distributed testing in which multiple machines are used to load test a single server.

It's critical that all machines are connected to the same network and have the same JMeter version. In distributed testing, one machine is designated as the master, while the others are designated as slaves.

The procedure is as follows:

- Edit the JMeter.properties file on the master system and add the IP addresses of slave machines to the remote host column.
- Save the file and then reopen JMeter.
- Select Remote Start from the RUN menu in JMeter and enter the IP address of the system to be invoked.
- To start all the slave machines for testing, go to the RUN menu and select Remote Start all.

4. Explain the usage of Regular Expression in JMeter. Also discuss the difference between ‘contains’ and ‘matches’.

Pattern-based regular expressions are used to find and manipulate text in Jmeter. JMeter uses the pattern matching software Apache Jakarta ORO to parse regular expressions or patterns used throughout a JMeter test plan.

We can save a lot of time and have more flexibility when creating or improving a Test Plan by using regular expressions. When it's impossible or very difficult to predict an outcome, regular expressions provide an easy way to acquire information from sites. Regular Expressions are used to dynamically extract values from responses. These values can be kept for reporting purposes or used in a subsequent request. Both Pre-Processors and Post-Processors make use of regular expressions.

The distinction between contains and matches, as used on the Response Assertion test element, is worth mentioning.

- The term “contains” indicates that at least some of the target was matched by the regular expression. Consider the string ‘alphabet’ and the regular expression ‘ph.b.’. Because the regular expression matches the substring 'phabe', we can say that 'alphabet' contains 'ph.b.'.
- The term "matches" refers to the regular expression matching the entire target. As a result, 'alphabet' and 'al.*t' are "matched."

5. Discuss about Samplers and Thread Groups in JMeter.

- A Thread Group is a collection of threads that are all working on the same problem. Every JMeter test plan starts with this element. Multiple thread groups are provided, each of which can be set to imitate how users interact with the application, how the load is sustained, and for how long.
- JMeter's samplers allow JMeter to make various types of requests to a server. JMeter provides genuine queries to the web server under test using samplers. Except for Test Action, each sampler generates one or more sample results. The sample results can be viewed in several ways and have various features (success/failure, elapsed time, data quantity, etc.).

6. What are the different types of processors available in JMeter?

In JMeter, there are two types of processors: the Pre-Processor and the Post-Processor.

- **Pre-Processors** run before the main sampler and have the ability to change the scope of the sampler. PreProcessors are JMeter elements that are used to perform actions prior to the execution of sampler requests in a test scenario. PreProcessors can be used for a variety of performance testing tasks, such as retrieving data from a database, setting a timeout between sampler operations, or generating test data.
- **Post Processors** run after the main sampler and apply to all samplers in the same Test Plan scope. They're useful for extracting fields from server responses and storing them in variables. The Post Processors are test plan elements that are used to do certain activities after a sampler request has been processed. These post processors are typically used to extract specific data from a sampling request's response, for example, we can extract the value of session variables from an HTTP request and send the value of the session variable to subsequent requests.

7. Mention the order of execution of JMeter test plan elements.

The execution order of the test plan elements is -

- **Configuration elements**- The Configuration Element allows us to declare variables that Samplers can utilise to access data.
- **Pre-processors**- PreProcessors are JMeter elements that are used to perform actions prior to the execution of sampler queries in a test scenario.
- **Timers**- A JMeter Timer is an inbuilt JMeter plug-in that can assist you in timing out your sampler requests. Many testers refer to this pause in requests as "think time," as it simulates the time spent by real users between steps in a real-world scenario.
- **Samplers**- JMeter uses samplers to make several types of queries to a server. JMeter provides genuine queries to the web server under test using samplers. Except for Test Action, each sampler delivers one or maybe more sample results.
- **Post-processors**- After the sampler, a Post-Processor element is executed and can be used for the post-condition response. If a Preprocessor is included in a test plan, it will be run after the response has been received. A Post-Processor is frequently used to extract information from a response to a specific action.
- **Assertions**- In JMeter, assertions are used to verify the response to a request that you have given to the server. The assertion is the process of comparing the expected and actual results of a request at runtime.
- **Listeners**- A listener is a component that displays the sample results. The output can be displayed in the form of a tree, tables, graphs, or simply sent to a log file.

8. Explain configuration elements in JMeter.

JMeter's config components are used to customise or change the sampler queries sent to the server. These components are inserted at the same level as or higher than the samplers we want to configure. A Sampler and a configuration element work in conjunction. Configuration elements can be used to set defaults and variables for subsequent usage by samplers. These elements are processed at the start of the scope before any samplers in the same scope.

9. What are the various data parameterization options in JMeter?

Data parameterization allows test scripts to be reused by removing the need for hardcoding values for the same request with varied parameters. In a single test script, parameterization is the process of establishing distinct datasets for different users. For instance, in the same script, executing numerous users with different credentials. As a result, it's one of the most important components of creating performance testing.

The data parameterization provided by JMeter is listed below:

- **Config of a CSV Data Set** - The CSV Data Set Config reads all values from a CSV file, stores them in variables, and uses them as Test Data during execution. The "CSV Data Set Config" allows you to save unique user data such as names, emails, and passwords in CSV files as an external data source. JMeter can read the CSV file line by line with the help of this config element, and then use split parameters to assign different values to separate threads.
- **Variables that are defined by the user** - User-defined variables (UDV) are used to define specific variables that can hold values that you need in several places. UserDefinedVariables differ from CSVDataSetConfig in that CSV can hold hundreds of thousands of values, whilst UDV is only suited for tiny data sets.

You can also specify variable/user values in UDV. This could be useful in instances where two users should use the same value and the next two users should use a different value. Also, if you declare a variable at the Test Plan/Thread Group level and then have the same variable with a different value inside one of two thread groups or a specific sampler. In this scenario, the local variable will override the global variable.

10. What is assertions in JMeter?

In JMeter, assertions are used to validate the response to a request that you have given to the server. The assertion is the process of comparing the expected and actual results of a request at runtime. If you want to apply assertion to a certain Sampler, make it a child of that Sampler. By adding "Assertion Listener" to the Thread Group, you can see assertion outcomes. Other listeners will be able to see failed assertions as well.

Some common assertions in JMeter are:

- **Response Assertion:** Response Assertion is used for combining and comparing pattern strings against one or more server response values.
- **Size Assertion:** Size Assertion is used to check whether the server response includes the expected number of bytes.
- **Duration Assertion:** The Duration Assertion is used to check whether or not a server response is received within a defined time limit. If the response takes longer than the specified time, the sample request will be marked as unsuccessful.
- **XML Assertion:** XML Assertion is used to check whether the server response data contains a valid XML document.
- **HTML Assertion:** HTML Assertion is used to ensure that the response contains proper HTML syntax and that Jtidy is not being used (HTML Syntax Checker). If the HTML syntax response is incorrect, the test will fail.

11. What is the maximum number of threads that should be allowed on a single system?

It depends on your system's hardware setup, which includes a processor, JVM, and allotted memory -Xmx, among other things. The number of components in your test plan, such as the number of config items or processors, as well as whether you are using GUI or Non-GUI Mode, have an impact on thread count.

12. Discuss Gaussian and Poisson Timers.

JMeter sends requests immediately, with no latency between samplers or requests. Your server will be overburdened if you perform load/stress testing immediately. Then it won't be able to provide you with realistic findings and won't be able to mimic real-world user traffic. JMeter Timers are the answer to all of these issues. To apply a wait between each sampler/request, a timer element can be added to a test plan.

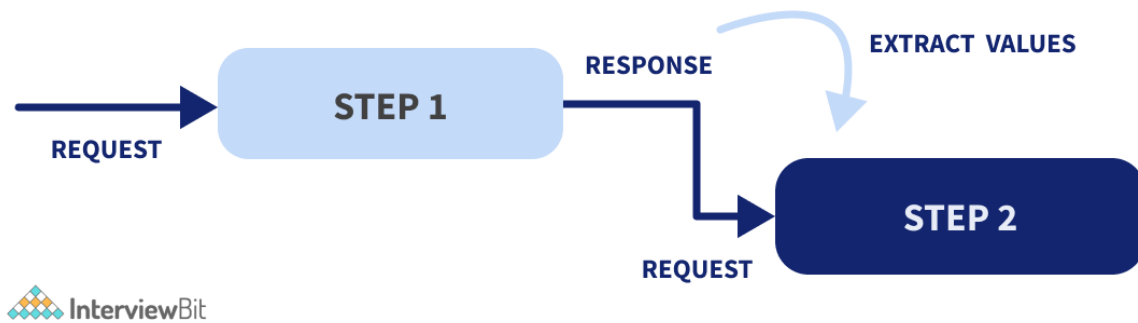
Both Gaussian and Poisson Timers are based on a mathematical formula with a constant delay and offset.

- **JMeter Gaussian Random Timer** is used to produce and add a random delay before executing a sampler, as the name implies. Each user request is delayed for a random period of time using the Gaussian Random Timer element. Based on the Gaussian curve distribution, it has a random deviation around the Constant Delay Offset. The sum of the Gaussian distributed value (with mean 0.0 and standard deviation 1.0) times the deviation value you choose and the offset value is the delay (think) time.
- **JMeter Poisson Random Timer** is also used to produce and add a random delay before executing a sampler. The Poisson Distribution Function is used to create this timer. The total of the Poisson distributed value multiplied by the defined lambda value and the offset value is the delay (think) time.

13. In JMeter, explain the purpose of correlation.

Correlation is the process of taking values from a server response and then storing them in a variable that will be used in subsequent requests. Correlation is the way of acquiring some value from one step's response and transferring it to another step's request. It catches and saves the server's dynamic answer before passing it on to subsequent requests. When a response returns different data for each iterating request, it is termed dynamic, and this can sometimes affect subsequent requests. Correlation is a critical step in the performance load test scripting process since if it isn't handled properly, the script will become ineffective. One of the most fundamental components of scripting is a correlation. It retrieves dynamic data from previous requests and posts it to new requests.

For example, if you need to test any login feature and need to use the session ID or the cookie ID, you can obtain the values from the login page's GET response and use them dynamically for login in a POST request.



14. What are the various kinds of listeners?

Performance testing is examining server responses in multiple formats before delivering them to the client. As a sampler component of JMeter is implemented, listeners give a graphical representation of data gathered by JMeter regarding those test cases. The user can view sampler results in the form of tables, graphs, trees, or plain text in a log file.

Listeners can be changed at any point during the test, including within the test plan. JMeter provides roughly 15 listeners, but the most commonly used ones are table, tree, and graph. In short, Listeners are used to storing the results of load testing execution in various formats, such as a table, tree, graph or any other appropriate format that may be given to the client. JMeter comes with a variety of built-in listeners, and many more can be added through plugins, depending on your needs. The following are a few of the inbuilt listeners: View Results in Table, Graph Results, Aggregate Report, Aggregate Graph, Response Time Graph and Assertion Results.

15. What exactly is a workbench?

A workbench is a storage location for adding some components to the test plan if necessary. Workbench components are not immediately saved with the test plan. They must be saved as test fragments individually. The HTTP(s) Test script recorder is a key component of the Workbench, as it allows you to record HTTPS requests and then apply a load on +9859 them to assess response time.

16. Discuss the working of the Test Script Recorder.

The HTTP(s) Test Script Recorder records all of your application's HTTP(s) queries to the server. It will only collect HTTP(s) requests, as the name implies. You can also use this recorder's grouping feature to structure your requests so that comparable requests are kept together. As needed, add URL patterns to the include and exclude lists. In order for JMeter to work, some configurations must be done there.

The following are the steps to capture HTTPS traffic: -

- HTTP(s) Test script recorder is to be added to WorkBench.
- To start the proxy server, type in the port number.
- Select "Workbench" as the target, or include a Recording Controller in the test plan and use the same target to save all recordings.
- The proxy server should now be running.
- Manually configure your browser's proxy settings to the same port number as the test script recorder.

17. What protocols are supported by JMeter?

JMeter supports a number of standard protocols, including:

- Web: HTTP/HTTPS
- Web Services: SOAP / XML-RPC
- Database via JDBC drivers
- Directory: LDAP
- Messaging Oriented service via JMS
- Service: POP3, IMAP, SMTP
- FTP (File Transfer Protocol) Service

18. Explain the JMeter variable and function syntax.

Variables and functions are utilised in JMeter, just as they are in any other programming language, to make scripts reusable (since a function can be called again and again without having the need to write the same code again and again). Variables can be used to hold any value and then referenced anywhere in your code, just like any other programming language. So, with JMeter, it may be your server name, path, port number, or whatever else you want to save in a variable and utilise in multiple places afterwards.

JMeter has a number of built-in routines that can be used to test various situations. If you're accessing an application on your machine IP rather than an external URL, for example, you can use the machine IP function to get your IP address. JMeter's Function Dialogue may be found in the settings menu, where you can examine the definition and function string.

The syntax for calling a variable and function in your script is shown below.

- Variable – {varName}.
Here, varName denotes the name of the variable.
Example – {name}
- Function – {_functionName}.
Here, functionName denotes the name of the function.
Example – {_counter()}, {_threadNum} etc.

JMeter Interview Questions for Experienced

19. Is it possible to use JMeter to record actions using a mobile device? If so, how would you go about doing it?

Yes, JMeter can also record HTTP or HTTPS requests sent to the server through your mobile application. Mobile and JMeter must be connected to the same network.

The following is the needed configuration:

- In JMeter, set your proxy server to run on a specific port.
- Set up the proxy on your phone's wifi settings and use the same port number as the recorder.
- On your mobile device, install the Root CA certificate.
- From your mobile, send server requests and watch them get captured by the selected controller.

20. Why is running JMeter in Non-GUI mode recommended?

JMeter tests can be conducted in GUI or non-GUI modes. Because the AWT (Abstract Window Toolkit) event thread can kill the tests in high load conditions, it is highly advised to conduct the load test in Non-GUI mode. JMeter's GUI mode is ideal for adding and updating new configuration components, thread groups, and samplers, as well as viewing a variety of different listeners for troubleshooting. The GUI option, on the other hand, has a constraint that slows down CPU consumption while running the recorded script. JMeter performance suffers when several listeners are used in a script. In order to avoid this, the script should be run in non-GUI mode. In non-GUI mode, there is a benefit to driving more requests per second out of JMeter.

JMeter supports a number of non-GUI modes, including:

- Command-line
- ANT plugin
- MAVEN plugin
- Jenkins

21. How do you perform spike testing with JMeter?

Spike Testing is a type of performance assessment that examines how the system behaves when the load is significantly increased on the system. Using the Synchronizing Timer in JMeter, you may perform spike testing. The threads are clogged by synchronising the timer until a certain number of threads are blocked, then releasing them all at once, resulting in a significant sudden load.

JMeter offers you two options to carry out spike testing:

- **Use the JMeter Ultimate Thread Group:** This strategy will necessitate advance planning. Using the Ultimate Thread Group, the tester can perform many ramp-ups and ramp-downs in a single test. These could be gradual or more abrupt. For our purposes, you could introduce a spike during the test, for example, after 10 minutes of slow ramp-up and then stable load, ramp up to 10X in 10 seconds, then ramp down to the previous load level after 1 minute. The disadvantage of this method is that you won't be able to regulate concurrency during the test run. You won't be able to predict when the spike will occur.
- **Use JMeter Properties:** This method will provide you with the most choice and flexibility. Determine how many threads you'll run at the start of the test, as well as when you'll want to create spikes dynamically. Use JMeter Properties for the number of threads, then use Real-Time Control to change the Properties while the test is executing.

22. Is it possible for you to use JMeter to run Selenium scripts? If so, how would you go about doing it?

Yes, you can run Selenium scripts in JMeter in order to get an indication of how well they perform.

- It can be done in two ways. You can either utilise JUnit libraries to create Selenium scripts and save them as Jars, which you can then add to the JMeter directory. Then, in your test plan, add the JUnit sampler and import the Jar file.
- Or else, you can place the Webdriver sampler plugin in the JMeter ext folder. JMeter should be restarted. In the Webdriver sampler, write your Selenium code and then run it to observe how it performs.

23. In JMeter, how do you handle sessions and cookies?

In JMeter, sessions and cookies may be controlled via config components like HTTP Cache Manager, which allows you to clear cookies after each iteration and add user-defined cookies. The HTTP Cache Manager assists you in cleaning the cache after every iteration as needed in load tests, as well as limiting the number of objects that can be cached. The HTTP sampler can be configured with both of these config elements.

24. What are the crucial steps in the JDBC (Java Database Connectivity) request testing process?

JDBC requests are used to create a connection with databases and then to track query response times.

The following are important procedures to take when evaluating JDBC requests:

- Setting up the Config Element, JDBC Connection setup, and adding the Database URL and JDBC Driver Class according to the database. Add the name of the variable for this connection configuration so that it can be used in the sampler.
- JDBC Request should be added. Write your queries to the test using the same variable name as before.

25. What is BeanShell scripting?

BeanShell is one of the most advanced JMeter built-in components. BeanShell is a lightweight Java scripting language that JMeter uses to do some sophisticated tasks. With the help of coding, the BeanShell sampler can execute a variety of tasks. You can get the thread number, run the current sampler, collect the cookies, and so forth. It comprehends Java syntax and provides scripting features such as loose types, commands and method closures. BeanShell can be a wonderful solution to achieve your goals if your test case is peculiar and creating it using embedded JMeter components is nearly impossible. BeanShell entities in JMeter include accessibility both to internal JMeter APIs as well as any external classes packed into the JMeter classpath (make sure to drop appropriate jars into your JMeter installation's /lib/ext folder and place the relevant "import" statements at the top of your BeanShell scripts).

The following BeanShell-enabled components are available in all JMeter versions:

- **Beanshell Sampler:** A sampler that can operate independently.
- **Beanshell PreProcessor:** A sampler pre-processor that operates before the sampler and can be used to establish prerequisites (i.e. to generate some input).
- **Beanshell PostProcessor:** A post-processor that runs after the sampler and can be used for cleanup or recovery.
- **Beanshell Assertion:** A powerful assertion that allows you to use the JMeter API completely. Java conditional logic can be used to control the outcome of the assertion.
- **__Beanshell Function:** A JMeter function that permits custom BeanShell code to be executed while the sampler is running.

26. Write the code to write data stored in a JMeter variable to a CSV file.

To write data to a CSV file, use the lines of code below inside a BeanShell postprocessor.

```
dataString = vars.get("DataToBeWritten");
filePath = vars.get("DataFilePath");
// you should pass true if you want to append the data to an existing file
// if you wish to overwrite, then do not pass the second argument
FileWriter fstream = new FileWriter(filePath, true);
BufferedWriter out = new BufferedWriter(fstream);
out.write(dataString);
out.write(System.getProperty("line.separator"));
out.close();
fstream.close();
```

In the above code, the data is written to an existing file. The path of the existing file is stored in filePath, and the data in dataString is appended to the file.

27. What is a Root Certificate Authority (CA) certificate?

A certificate is required to authenticate HTTPS connections, which are established when the browser connects to the webserver. JMeter creates it for the purpose of intercepting SSL traffic and recording actions. This certificate must be installed on your mobile device in order to record actions via mobile. It is necessary to install the JMeter protected CA certificate in the browser before recording any secured web application with JMeter.

The HTTP(S) Test Script Recorder in JMeter can be used to record the protected online application. Certificates are used to authenticate the connection between the browser and the web server in secured (HTTPS) communications. The server presents the certificate to the browser when connecting through HTTPS. The browser verifies that the server certificate is signed by a Certificate Authority (CA) that is related to one of the browser's built-in root CAs. To intercept the HTTPS connection from the browser, JMeter must utilise its own certificate to enable encrypted connections. JMeter must effectively impersonate the target server.

28. What factors influence the number of threads that should be generated per system?

It is dependent on the system's hardware.

On a 2-3 GHz CPU, for example, 400-600 threads can be produced. It also depends on the elements included in your test strategy. The higher the number of processors and XML processing components, the higher the CPU burden and, as a result, the fewer threads. Load testing with multiple machines is recommended for large loads.

29. What are the most significant plugins that JMeter supports?

JMeter comes with a lot of functionality by default, but one of the best things about it is that its open-source software! As a result, any interested party can create additions to the system that will expand its capabilities and provide new features. Plugins are the name for these additions. Developers have already produced a wide range of useful plugins, which is fantastic and demonstrates the JMeter community's strength. JMeter supports a variety of plugins that aid in the generation of high-quality results.

The important plugins that are supported are listed below:

- **Plugins Manager:** The Plugin Manager is the most popular plugin, and for a legitimate reason: most users need to install the Plugin Manager before installing additional plugins. The most convenient approach to handle JMeter plugins is to use the Plugins Manager.
- **Custom Thread Groups:** Stepping Thread Group, Ultimate Thread Group, Concurrency Thread Group, Arrivals Thread Group, and Free-Form Arrivals Thread Group are the five thread group types added by the Custom Thread Groups plugin. These five thread groups provide you with a lot of possibilities for creating test run schedules.
- **Dummy Sampler:** The Dummy Sampler simulates request and response processing without actually running the requests. The sampler's fields define the request and response data. Debugging post-processors and extractors has never been easier.
- **PerfMon (Servers Performance Monitoring):** This plugin adds the PerfMon Servers Performance Monitoring listener to JMeter. This listener allows us to keep track of the servers' CPU, memory, swap, disc I/O, and network I/O.
- **Throughput Shaping Timer:** The following features are added to JMeter by this plugin: Throughput Shaping Timer, Special Property Processing, and Schedule Feedback Function. These features allow us to set a test throughput limit to guarantee that we don't go above our desired throughput.
- **Basic Graphs:** JMeter is extended with three listeners by the 3 Basic Graphs plugin: Active Threads Over Time, Response Times Over Time, and Transactions per Second. These listeners display information and KPIs regarding the load testing results in a graphic style, allowing you to examine the success of your performance tests and make more informed decisions about your website or app's next steps.

30. In JMeter, what are the different sorts of controllers?

In JMeter, controllers are used to regulating the flow of request execution. They allow you to specify the order in which requests are processed in a Thread. It allows you to choose "when" a user request is sent to a web server. They choose the sequence in which user requests are processed.

Some controllers used in JMeter are listed below:

- **Recording controller**- JMeter has the ability to record your testing steps; a recording controller is a container for these recordings.
- **IF controller**- IF Controller decides whether or not to run a batch of child samplers based on particular criteria.
- **While controller**- The JMeter While Controller basically runs child Samplers and keeps running until the condition in the condition field becomes false.
- **Transaction controller**- The Transaction Controller in JMeter is a useful tool for organising and determining how different pieces of your test will appear in a report. The Transaction Controller creates a second sample that measures the total time it takes to complete the nested test items.
- **Loop controller**- The Loop Controller causes the user request to be repeated a given number of times or to be repeated indefinitely.
- **Simple controller**- Simple Controller is a container for user requests.
- **Module controller**- Module Controller's purpose is to make JMeter more modular. Web applications, in general, are made up of small functional units (e.g., login, create account, logoff...). This capability can be saved as "modules" in Simple Controller. The Module Controller will determine which module is required to run.
- **Random Controller**-In each loop period, the Random Controller causes all user requests to be processed in a random sequence.
- **Interleave Controller**- Interleave Controller takes one of the user requests and makes it run in each loop of the thread.

31. How is Ultimate thread group different from other thread groups?

Multiple thread groups are provided, each of which can be set to imitate how users interact with the app, how the load is sustained, and for how long. The Arrivals Thread Group is the beginning of thread iteration. It defines the schedule of the load. The Concurrency Thread group is likewise appropriate for goal-oriented scenarios, but the purpose here is to manage the number of concurrent users over time. Freedom Arrivals Thread Group resembles the Arrivals Thread group, the difference being that this thread group does not have choices for ramp-up time and steps. The Stepping Thread Group is an older version of the Concurrency Thread Group that takes a little more configuration. The ultimate result is essentially the same; the main difference is in the ramp-up and ramp-down phases.

The ultimate thread group is highly customizable, and unlike the arrivals, freeform, and concurrency thread groups, it kills active threads when the set time runs out. As users exit the application tab or browser, this would convert into real-world user behaviour. Because the ramp-down is controlled by the thread group rather than the execution time of individual threads, the preview graph depicting the expected simultaneous users will be extremely accurate. In the case where one thread has completed execution, the thread group also starts a new thread to keep the thread count up.

Conclusion:

We are confident that this post on JMeter interview questions has greatly enriched your knowledge of JMeter topics. A solid mastery of all of the questions provided here will enable you to confidently crack any interview. Happy learning!

Links to More Interview Questions

[C Interview Questions](#)

[Php Interview Questions](#)

[C Sharp Interview Questions](#)

[Web Api Interview Questions](#)

[Hibernate Interview Questions](#)

[Node Js Interview Questions](#)

[Cpp Interview Questions](#)

[Oops Interview Questions](#)

[Devops Interview Questions](#)

[Machine Learning Interview Questions](#)

[Docker Interview Questions](#)

[Mysql Interview Questions](#)

[Css Interview Questions](#)

[Laravel Interview Questions](#)

[Asp Net Interview Questions](#)

[Django Interview Questions](#)

[Dot Net Interview Questions](#)

[Kubernetes Interview Questions](#)

[Operating System Interview Questions](#)

[React Native Interview Questions](#)

[Aws Interview Questions](#)

[Git Interview Questions](#)

[Java 8 Interview Questions](#)

[Mongodb Interview Questions](#)

[Dbms Interview Questions](#)

[Spring Boot Interview Questions](#)

[Power Bi Interview Questions](#)

[Pl Sql Interview Questions](#)

[Tableau Interview Questions](#)

[Linux Interview Questions](#)

[Ansible Interview Questions](#)

[Java Interview Questions](#)

[Jenkins Interview Questions](#)