Apache Tomcat Interview Questions and Answers

Apache Tomcat and the Apache httpd server, were created for two different tasks but sometimes We can confuse the functionality of each other just because they could solve the same problems.

### Table Of Contents

[1. Introduction](https://examples.javacodegeeks.com/enterprise-java/tomcat/apache-tomcat-interview-questions-answers/#toc100)

[2. General questions](https://examples.javacodegeeks.com/enterprise-java/tomcat/apache-tomcat-interview-questions-answers/#toc200)

[3. Tomcat internals](https://examples.javacodegeeks.com/enterprise-java/tomcat/apache-tomcat-interview-questions-answers/#toc300)

[4. Servlets](https://examples.javacodegeeks.com/enterprise-java/tomcat/apache-tomcat-interview-questions-answers/#toc400)

[5. Tomcat configuration files](https://examples.javacodegeeks.com/enterprise-java/tomcat/apache-tomcat-interview-questions-answers/#toc500)

[6. Tomcat connectors](https://examples.javacodegeeks.com/enterprise-java/tomcat/apache-tomcat-interview-questions-answers/#toc600)

[7. Tomcat and other servers](https://examples.javacodegeeks.com/enterprise-java/tomcat/apache-tomcat-interview-questions-answers/#toc700)

[8. Tomcat components](https://examples.javacodegeeks.com/enterprise-java/tomcat/apache-tomcat-interview-questions-answers/#toc800)

[9. Tomcat architecture](https://examples.javacodegeeks.com/enterprise-java/tomcat/apache-tomcat-interview-questions-answers/#toc900)

[10. Conclusion](https://examples.javacodegeeks.com/enterprise-java/tomcat/apache-tomcat-interview-questions-answers/#toc1000)

## 1. Introduction

In this article We are going to show Tomcat Server frequently Asked Questions in various Tomcat job Interviews. As a Tomcat administrator your task is to be in charge of all Tomcat common tasks like server maintenance, upgrades, manage users and add new features.

Top of Form

# Want to master Tomcat Server?

|  |
| --- |
| Subscribe to our newsletter and download the Apache Tomcat Cookbook right now!**In order to get you prepared for your Tomcat development needs, we have compiled numerous recipes to help you kick-start your projects. Besides reading them online you may download the eBook in PDF format!**   **Email address:** |

Bottom of Form

## 2. General questions

**1. What is Tomcat?**

Tomcat is a Java Servlet container and web server from the Apache software foundation. It can be used as standalone or it can be used behind traditional web servers such as Apache httpd.

Recent versions of Tomcat can serve static content as fast as httpd. A web server uses the request/response message exchange pattern to serve web pages. Tomcat provides Servlet, JSP technologies and static content as well.

**2. Who is responsible of Tomcat?**  
The Apache Software Foundation. The Apache Software Foundation is an umbrella organization that looks after a number of Open Source projects.

Jakarta is the group name for the Java based projects of the Apache Software foundation.

Tomcat is a Web Server that handles server side Java (in the form of Servlets and JSPs), and it’s a part of the Apache Jakarta project group. Tomcat is the “reference” implementation of the Servlet and JSP standards – in other words, if it runs under Tomcat, it should run under any compliant Servlet / JSP container.

**3. What do you know about Tomcat history?**

Tomcat started off as a servlet reference implementation by James Duncan Davidson, a software architect at Sun Microsystems. He later helped make the project open source and played a key role in its donation by Sun Microsystems to the Apache Software Foundation. The Apache Ant software build automation tool was developed as a side-effect of the creation of Tomcat as an open source project.

**4. Name some Tomcat features**

Tomcat 7.x implements the Servlet 3.0 and JSP 2.2 specifications. It requires Java version 1.6. Tomcat 8.x implements the Servlet 3.1 and JSP 2.4 Specifications. Tomcat 8.5.x is intended to replace 8.0.x and includes new features pulled forward from Tomcat 9.0.x. Tomcat 8.5 is designed to run on Java SE 7 and later.

## 3. Tomcat internals

**5. What is the Tomcat default port?**

The default port for Tomcat is 8080. You can change the default port by editing the file server.xml under the conf folder in the Tomcat installed directory. Change the property Connector port=”8080″ to the desired port and restart Tomcat so the changes can take effect.

**6. How to deploy War web applications in Tomcat**

You can drop the WAR file inside the webapps folder or use the Tomcat manager to deploy War files.

## What is Secure Sockets Layer (SSL)?

Secure Sockets Layer (SSL) is a standard security technology for establishing an encrypted link between a server and a client—typically a web server (website) and a browser, or a mail server and a mail client (e.g., Outlook).

SSL allows sensitive information such as credit card numbers, social security numbers, and login credentials to be transmitted securely.

**7. Can Tomcat use SSL?**

Yes, you need to make additional configurations to make Tomcat use SSL. In resume you need to do these tasks

* Generate Keystore
* Add a conector in server.xml
* Restart Tomcat

**8. How do I override the default home page loaded by Tomcat?**

Inside $TOMCAT\_HOME/conf/web.xml there is a section called <welcome-file-list> and it looks like this:

welcome-file-list

|  |  |
| --- | --- |
| 1 | <welcome-file-list> |
| 2 | <welcome-file>index.html</welcome-file> |

|  |  |
| --- | --- |
| 3 | <welcome-file>index.htm</welcome-file> |
| 4 | <welcome-file>index.jsp</welcome-file> |

|  |  |
| --- | --- |
| 5 | </welcome-file-list> |

The default servlet attempts to load the index.\* files in the order listed. You may easily override the index.jsp file by creating an index.html file at $TOMCAT\_HOME/webapps/ROOT.

**9. What services are provided by Tomcat?**

Tomcat server provides a host of services which are not provided by normal web servers like Apache Web Server. Those are:

* Servlet Life cycle
* Handle Web Requests
* Database connection pooling
* Clustering
* High availability

## 4. Servlets

**10. What is a servlet container?**

The servlet container is the component of a web server that interacts with Java servlets. The servlet container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access-rights.

The servlet container handles requests to servlets, JavaServer Pages (JSP) files, and other types of files that include server-side code. The Web container creates servlet instances, loads and unloads servlets, creates and manages request and response objects, and performs other servlet-management tasks.

The servlet container implements the web component contract of the Java EE architecture, specifying a runtime environment for web components that includes security, concurrency, lifecycle management, transaction, deployment, and other services.

**11. What is the servlet container life cycle?**

1 Servlet life cycle

A servlet life cycle can be defined as the entire process from its creation till the destruction.

Life cycle steps followed by a servlet

* The servlet receives a request from a client through one of its connectors then is initialized by calling the init() method.
* The servlet calls service() method to process a client’s request and send the response.
* The servlet is terminated by calling the destroy() method.
* The servlet is garbage collected by the garbage collector of the JVM.

## 5. Tomcat configuration files

**12. What is the server server.xml configuration file?**

The server.xml file is Tomcat main configuration file, and it is responsible for specifying Tomcat configuration on startup.

**13. What is web.xml configuration file?**

The web.xml file is derived from the Servlet specification, and contains information used to deploy and configure the components of your web applications.

**14. What is Tomcat-users.xml configuration file?**

It is where the Tomcat users are defined and it is located in the conf folder of the Tomcat server root.

**15. Where do you configure a database connection pool in Tomcat server?**

The Configure pool is in the context.xml inside the conf folder of tomcat.

## 6. Tomcat connectors

**16. What is a connector and why it is used in Tomcat?**

The Apache Tomcat Connectors project is a part of the Tomcat project and provides web server plug-ins to connect web servers with Tomcat and other back-ends.  
The supported web servers are:

* The Apache HTTP Server with a plugin named mod\_jk.
* Microsoft IIS with a plugin named ISAPI redirector.
* The iPlanet Web Server with a plugin named NSAPI redirector.

## 7. Tomcat and other servers

**17. What is the difference between Tomcat and an Application server**

Tomcat is a servlet container that supports servlets and the JSP technology. An Application server supports many other Java EE technologies.

**18.What Application server is based on Tomcat ?**

TomEE.

**19. What is TomEE?**

Apache TomEE (pronounced “Tommy”) is the Java Enterprise Edition of Apache Tomcat (Tomcat + Java EE = TomEE) that combines several Java enterprise projects including Apache OpenEJB, Apache OpenWebBeans, Apache OpenJPA, Apache MyFaces and others.

## 8. Tomcat components

**20. What is Jasper?**

Jasper is Tomcat’s JSP Engine. Jasper parses JSP files to compile them into Java code as servlets. At runtime, Jasper detects changes to JSP files and recompiles them. Jasper is the Java Server pages handler in Tomcat; internally, it deals with any compiling that’s neccessary, and converts JSPs into Servlets for Catalina to handle.

**21. What is Catalina ?**

Catalina is Tomcat’s servlet container. Catalina implements specifications for servlet and JavaServer Pages. Catalina is the Java Engine (JRE / JVM) that’s built into Tomcat and provides an environment in which Servlets can be run.

**22. What is Coyote ?**

Coyote is a Connector component for Tomcat that supports the HTTP 1.1 protocol as a web server. This allows Catalina, nominally a Java Servlet or JSP container, to also act as a plain web server that serves local files as HTTP documents.

Coyote listens for incoming connections to the server on a specific TCP port and forwards the request to the Tomcat Engine to process the request and send back a response to the requesting client.  
Coyote is the HTTP connector that’s built into Tomcat and provides Tomcat with an interface that browsers can connect to.

**23. What is a Tomcat cluster ?**

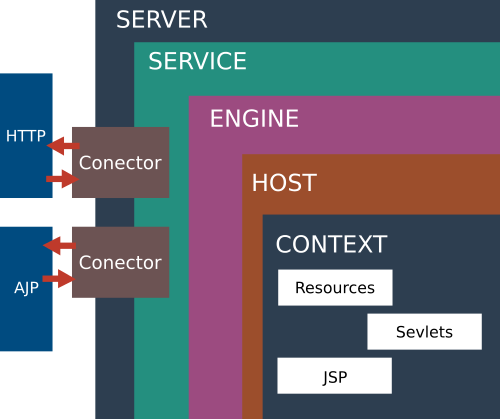
This component is used to manage large applications. It is used for load balancing and can be achieved through many techniques. Apache Tomcat cluster is used to manage more traffic. It provides multiples instances of the Tomcat server with its content balanced between these instances.

**24. What is a Tomcat High availability ?**

A high-availability feature has been added to facilitate the scheduling of system upgrades without affecting the live environment. This is done by dispatching live traffic requests to a temporary server on a different port while the main server is upgraded on the main port. It is very useful in handling user requests on high-traffic web applications.

## 9. Tomcat architecture

**25. What is the architecture of a Tomcat server?**

[](http://examples.javacodegeeks.com/wp-content/uploads/2016/10/2-Tomcat-Architecture.png)

2 Tomcat Architecture

Tomcat architecture

|  |  |
| --- | --- |
| 01 | <server> |
| 02 | <service> |

|  |  |
| --- | --- |
| 03 | <conector/> |
| 04 | <engine> |

|  |  |
| --- | --- |
| 05 | <host> |
| 06 | <context> |

|  |  |
| --- | --- |
| 07 | </context> |
| 08 | </host> |

|  |  |
| --- | --- |
| 09 | </engine> |
| 10 | </service> |

|  |  |
| --- | --- |
| 11 | </server> |

**26. What is the server?**

A Server element represents the entire Catalina servlet container. Therefore, it must be the single outermost element in the conf/server.xml configuration file. Its attributes represent the characteristics of the servlet container as a whole.

**27. What is the Service?**

A Service element represents the combination of one or more Connector components that share a single Engine component for processing incoming requests. One or more Service elements may be nested inside a Server element.

**28. What is the engine ?**

The Engine element represents the entire request processing machinery associated with a particular Catalina Service. It receives and processes all requests from one or more Connectors, and returns the completed response to the Connector for ultimate transmission back to the client.

Exactly one Engine element MUST be nested inside a Service element, following all of the corresponding Connector elements associated with this Service.

**29. What is the host ?**

The Host element represents a virtual host, which is an association of a network name for a server.

**30. What is the Connector ?**

A Connector represents an end point in which requests are received.

**31. What is the Context ?**

The Context element represents a web application, which is run within a particular virtual host. Each web application is based on a Web Application Archive (WAR) file, or a corresponding directory containing the corresponding unpacked contents, as described in the Servlet Specification.

## 10. Conclusion

Those are some common questions that you can face in a Tomcat job interview. Please remember, only theory doesn’t has much value without the practice, so, if you are going to a Tomcat interview make sure you understand how it works.

Top of Form

Bottom of Form