

# Application Server Implementation

# What is a web server?

- ▶ A Web Server is a computer program or a computer that runs the application. It is the main feature of accepting HTTP requests from clients and deliver webpage and then serving back HTTP responses. It could also be determined as a virtual machine program. This type of delivery consists of HTML documents or additional content like style sheets and JavaScript.

# What is Application Server?

- ▶ An application server is a sever type that helps you to host applications. It provides both facilities to create a web app and a server environment to run them.
- ▶ This type of application server clients can include a webserver, GUIs running on a PC, or even other application servers. All the information is traveling back and forward between an application server and its client. It is also not restricted to just display mark-up.
- ▶ The information is program logic, which takes the form of data and method and not static HTML. The client allows you to employ the exposed business logic whenever they want.
- ▶ For example, application servers, like the Apache Tomcat, power the interactive parts of a website, which may appear depends on the context of the request.

# Web Server Vs Application Server

- ▶ Web Server is responsible for accepting HTTP requests from clients and serving back that HTTP responses whereas Application server exposes business logic to the clients, which generates dynamic content.
- ▶ Web servers are used for producing static or dynamic, hypertext documents and Application servers use for text document generation for the computation on provided data.
- ▶ Web server consumes fewer resources like CPU memory compared with the application server while the application server utilizes more resources.
- ▶ The web server supports HTTP/s Protocol but the application server supports HTTP/s and RPC/RMI protocols.
- ▶ Web server provides an environment to run a web application but the application server gives an environment to run the web with enterprise applications.

# Features of Web Server

Here are important features of a web server:

- ▶ Handles HTTP Protocol (static contents)
- ▶ No Server-side Programming.
- ▶ Support web-Based Applications (JSP, Servlets, PHP, HTML, etc.)
- ▶ Not support Database Connection Pooling.
- ▶ Not provide EJB support.



# Features of Application Server

Here are important features of a web server:

- ▶ Serves dynamic business logic.
- ▶ It helps you to manage backend logic like calculations, database, processing, etc.
- ▶ It helps you to deploy applications, dependency injection, security, etc. database pooling, and EJB.
- ▶ The superior server of Web Server.

# Differences between Web and Application Server:

## Web Server

- ▶ Web Server is a computer program or a computer that runs the application.
- ▶ It is responsible for accepting HTTP requests from clients and serving back that HTTP responses.
- ▶ Subset of the application server.
- ▶ Web servers are used for producing produce static or dynamic, hypertext documents.

## Application Server

- ▶ An application server is a sever type that helps you to host applications.
- ▶ It exposes business logic to the clients, which generates dynamic content.
- ▶ Superset of a web server.
- ▶ Application servers use for text document generation for the computation on provided data.

## Web Server

- ▶ Web servers need a web browser to display the result of the HTTP request. This is commonly known as an HTTP response.
- ▶ It provides an environment for server-side programs to execute and produce HTTP response in results.
- ▶ Web servers are accessed using HTTP request and HTTP protocols.
- ▶ The result is a hypertext document storing the information which is displayed to the user on a web browser.

## Web Application

- ▶ The client-side application is need to continuous data exchange between the application server and client application.
- ▶ The application server offers an ambiance to expose the functionality of the software installed at the server-side to the clients.
- ▶ Application servers are accessed using APIs.
- ▶ The result is XML, JSON, and HTML, etc. files that contain required data and can serve a special purpose depending upon the user's needs.



## Web Server

- ▶ The web server does not support multithreading.
- ▶ Web servers are primarily designed to serve HTTP content.
- ▶ It provides an environment to run a web application.

## Web Application

- ▶ Application server assists multithreading and distributed transactions.
- ▶ Application server can also serve HTTP content. However, it also assists protocol like RMI/RPC.
- ▶ The application server gives an environment to run the web with enterprise applications.

## Web Server

- ▶ Web servers need a web browser to display the result of the HTTP request. This is commonly known as an HTTP response.
- ▶ It provides an environment for server-side programs to execute and produce HTTP response in results.
- ▶ Web servers are accessed using HTTP request and HTTP protocols.
- ▶ The result is a hypertext document storing the information which is displayed to the user on a web browser.

## Web Application

- ▶ The client-side application is need to continuous data exchange between the application server and client application.
- ▶ The application server offers an ambiance to expose the functionality of the software installed at the server-side to the clients.
- ▶ Application servers are accessed using APIs.
- ▶ The result is XML, JSON, and HTML, etc. files that contain required data and can serve a special purpose depending upon the user's needs.

# What is Tomcat Web Server

- ▶ It is one of the most popular Open-source Web Server and Servlet containers that run based on the Java Code. Tomcat is not a full application server as it provides only the webserver and Servlet Container, but these features can alone adequate for the application production.
- ▶ It is a widely used Application server; nearly 60% of the people in the world use the Tomcat as its application Server. This is mainly because of the lightweight and high efficiency of the Tomcat Web Server.

# History of Tomcat

- ▶ The development of the Tomcat started in 1998, just 4 years after the introduction of Java. It was first developed by Sun Microsystems, then its code was given to Apache Software Foundation and became Open Source Software.
- ▶ After it became open source many volunteers developed or updated the base code and created top-level Web Server software to load java based web pages and apps in 2005.

# Difference between the Tomcat and Apache Web Server

## Tomcat Web Server

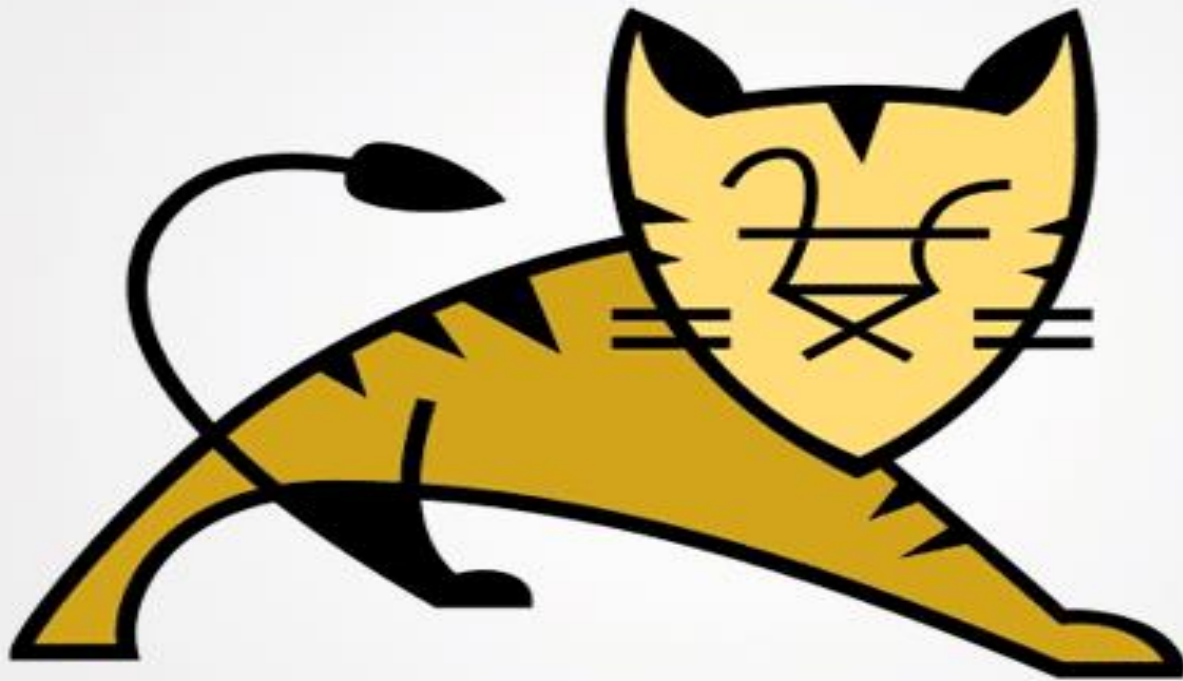
- ▶ It is a web container that allows the users to run Servlet and Web-Java Server pages
- ▶ It can handle both static and dynamic pages. Static pages are coded through HTML and Dynamic pages used JAVA codes.
- ▶ It can be used only for hosting JAVA-based code.
- ▶ It is coded by pure Java language.

## Apache Web Server

- ▶ It is a software developed to create Web-servers. It can host HTTP-based web servers.
- ▶ This also handles both types of pages, where HTML is used for Static pages and PHP or other languages are used for Dynamic pages using the add-on modules.
- ▶ It can be used to host applications or websites in any language.
- ▶ It is only coded through C language.



# Configure Apache Tomcat



**Apache**  
TOMCAT

# What is apache tomcat

- ▶ **Apache Tomcat** is an open-source, lightweight, powerful and widely-used web server developed and maintained by **Apache Foundation**. It is an implementation of the **Java Servlet**, **JavaServer Pages (JSP)**, **Java Expression Language (EL)** and **Java WebSocket** technologies, and provides a pure Java HTTP server to run Java web-based applications.

# Install java and tomcat

- ▶ Download java

<https://www.oracle.com/in/java/technologies/javase/jdk11-archive-downloads.html>

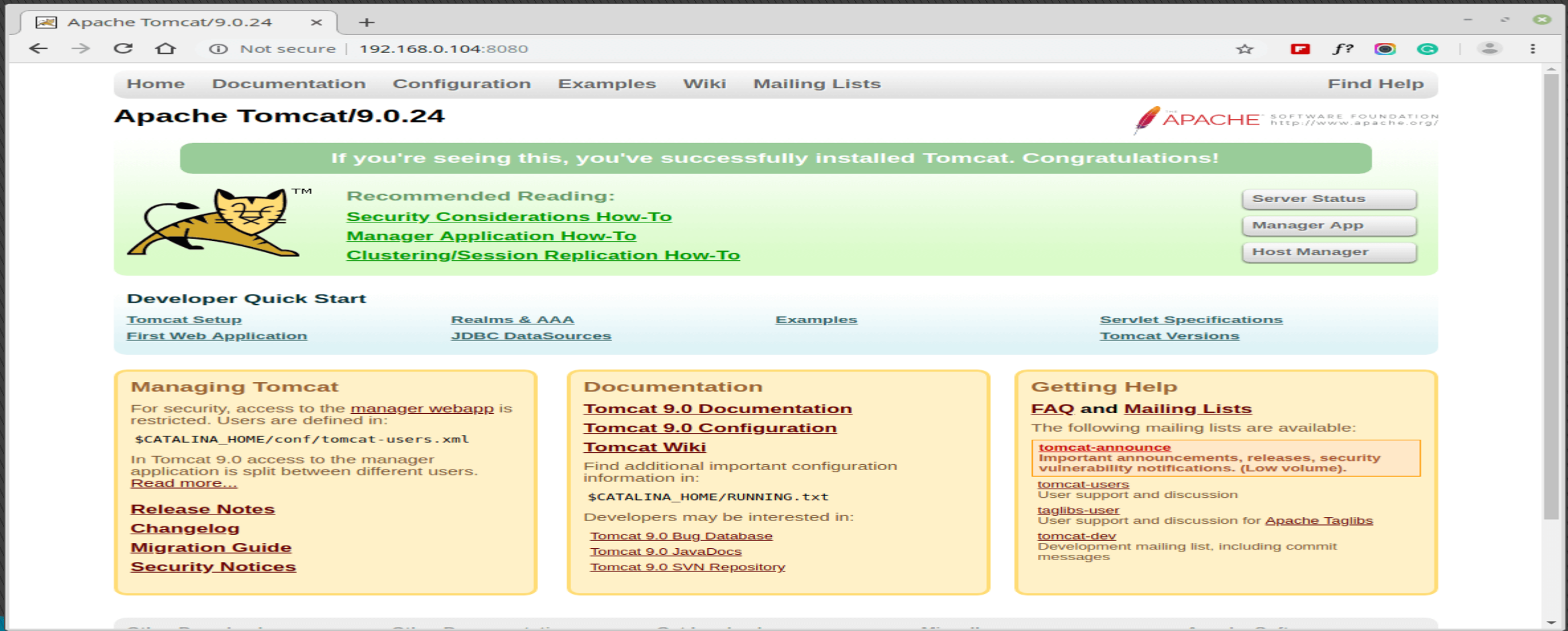
- ▶ Download tomcat

<https://tomcat.apache.org/download-90.cgi>

# Access Apache Tomcat Web Interface

- ▶ Now that you have installed, configured and started Tomcat as a service, you can test the installation by trying to accessing the web interface using the URL.
- ▶ `http://localhost:8080`
- ▶ OR
- ▶ `http://SERVER_IP:8080`
- ▶ Once you see the page shown in the screenshot, you've successfully installed Tomcat.

# Open using web browser




The screenshot shows a web browser window with the Apache Tomcat/9.0.24 homepage. The browser's address bar shows the URL `192.168.0.104:8080`. The page features a navigation bar with links to Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists, along with a Find Help button. A green banner congratulates the user for successfully installing Tomcat. Below this, there is a section for Recommended Reading with links to Security Considerations, Manager Application, and Clustering/Session Replication. To the right of these links are buttons for Server Status, Manager App, and Host Manager. The page is divided into three main columns: Developer Quick Start, Documentation, and Getting Help. The Developer Quick Start column includes links for Tomcat Setup, First Web Application, Realms & AAA, JDBC DataSources, and Examples. The Documentation column includes links for Tomcat 9.0 Documentation, Tomcat 9.0 Configuration, and Tomcat Wiki. The Getting Help column includes links for FAQ and Mailing Lists, and a list of available mailing lists.

Apache Tomcat/9.0.24

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 Recommended Reading:

- [Security Considerations How-To](#)
- [Manager Application How-To](#)
- [Clustering/Session Replication How-To](#)

Server Status  
Manager App  
Host Manager

Developer Quick Start

- [Tomcat Setup](#)
- [First Web Application](#)
- [Realms & AAA](#)
- [JDBC DataSources](#)
- [Examples](#)
- [Servlet Specifications](#)
- [Tomcat Versions](#)

Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 9.0 access to the manager application is split between different users.  
[Read more...](#)

[Release Notes](#)  
[Changelog](#)  
[Migration Guide](#)  
[Security Notices](#)

Documentation

[Tomcat 9.0 Documentation](#)  
[Tomcat 9.0 Configuration](#)  
[Tomcat Wiki](#)

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

- [Tomcat 9.0 Bug Database](#)
- [Tomcat 9.0 JavaDocs](#)
- [Tomcat 9.0 SVN Repository](#)

Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

- [tomcat-announce](#)  
Important announcements, releases, security vulnerability notifications. (Low volume).
- [tomcat-users](#)  
User support and discussion
- [taglibs-user](#)  
User support and discussion for [Apache Taglibs](#)
- [tomcat-dev](#)  
Development mailing list, including commit messages