**Introduction to Apache Tomcat**

If you are familiar with website, web pages, webapp, then http protocol must not be new to you, if you want to provide any web services, maybe if you want to provide a static content using html, maybe you want to send data from server to client servers, you need HTTP protocol to fulfil that task.

But to make your http server dynamic, you will need servlets because we use HTTP only to send data but to fetch a dynamic data, you need a servlet. So, we will have HTTP server and have a container where we will run our servlet.

So, when you combine both HTTP server and combine your servlet container, they both combined to give you what is called Tomcat.

Tomcat is not the only server available in the market as you must have known by now, we have Jetty also there appservers as well.

So what is Tomcat, it actually the server and the servlet container, some people do mix it up by separating Tomcat to be either servlet or HTTP server, but the actual truth is that Tomcat is both.

Apache Tomcat is open-source web server software for Java programming that is developed and maintains by the Apache software foundation. The initial idea of Apache tomcat software was to host and deploy the Java servlet that is the server-side Java code that manages HTTP results from client application build using Java.

It acts as a web server rather than a full-fledged application server that includes data persistence and load balancing capabilities. Apache Tomcat provides the basic feature of web server processing for the relevant servlets. It supports the java servlet lifecycle that are init(),service() and destroy() phases. It is the preferred web server software for Java implementations The latest stable release of a tomcat version 9.0.21 was released on June 7th, 2019.

Apache tomcat may be defined as a web server (that is also referred to as a web container/ servlet container), which processes the servlets, JSP’s (by internally converting your JSP’s to servlets internally) and also render JSP’s.

Note that a web server is different from that of an application server. Let us understand how these two differentiate from one another.

**Why is Tomcat said to be a Web Server but not an Application Server?**

1. When packaging an application, it can either be packaged as a .war or .ear Tomcat is categorized as a web server since it can handle only .war files.
2. A web server is often said to be a part of an application server since an application server exhibits all the features that a web server possesses and in addition to it also contains attributes such as load balancing, data persistence, messaging to name a few. However, vice versa does not apply.
3. Web servers can operate only over HTTP protocol, whereas application servers can operate over various CGI protocols and may also include HTTP protocol. Application servers are mostly used to handle large enterprise edition applications. Since tomcat operates only on the HTTP protocol, it falls under the webserver.

**Why we need to use Apache Tomcat?**

Web pages by itself are [static HTML files](https://www.educba.com/what-is-html/). Hence a client cannot interact with a static web page. In order to facilitate our web pages with dynamic capabilities, a web server is required. To be able to interface one’s application with the webserver, predefined API’s (Application Programming Interfaces) are provided. Servlet is one such API provided by the Java Platform Enterprise Edition designed to work along with web servers. Monitoring the server for incoming client requests is not the job of a servlet but that of a web server.

**How does Apache Tomcat work?**

Tomcat is widely used by web developers when working on [web application development](https://www.educba.com/what-is-web-application/). From a high-level perspective, apache tomcat is responsible to provide a run-time environment for the servlets. It provides an environment in which one could run their java code.

On a more detailed aspect, tomcat is responsible for:

1. Listen to all incoming requests from clients.
2. Load the respective servlet classes using the servlet mappings (from web.xml file) to handle incoming client requests.
3. Execute the servlet class and.
4. Finally, unload the servlet class.

From the point the servlet class is loaded to the point it’s unloaded, the servlet is responsible for handling the client request by carrying out its various life cycle methods and providing the necessary response back to tomcat as JSP pages. Tomcat then returns the response back to the client by rendering the JSP.

**Advantages of Apache Tomcat**

1. The biggest advantage of apache tomcat is that it’s open source. There is no need to shell out money in order to use this software. One could easily download it over the internet and configure it and start working with it.
2. The Apache software foundation provides regular updates to make it compatible with other software versions and providing bug fixes thus making it easier for developers to use.
3. Tomcat supports SSL (Secure Socket Layer) and therefore can be configured [using an SSL certificate](https://www.educba.com/what-is-ssl-certificate/) to secure sensitive data by providing a secure connection.
4. Tomcat can also be configured to run multiple web applications on different ports. For example, it could be running three applications on 8080, 8081, 9090 port numbers. By default, Apache tomcat makes use of port number 8080.
5. It is also cross-platform compatible, that is can be used on Windows, Mac OS, Linux operating systems.
6. It is said to be lightweight. That is, it consumes less in terms of memory and resource utilization, thus allowing the application to run smoothly on most systems without specific system requirements.

**Conclusion**

Apache Tomcat is the most widely and commonly used software among web application developers today. Studies have claimed that more than 60% of java applications make use of apache tomcat.