

# **TOMCAT, JAKARTA PROJECT, JSP AND SERVLETS**

**Eduardo Gallego Chillón – 2º DAW  
IES FRANCISCO AYALA (Granada)**

# Index

<b>1. Tomcat.....</b>	<b>2</b>
<b>1.1. Environment.....</b>	<b>2</b>
<b>1.2. Directory structure.....</b>	<b>2</b>
<b>1.3. Features.....</b>	<b>2</b>
<b>2. Jakarta project.....</b>	<b>3</b>
<b>3. JSP (JavaServer Pages).....</b>	<b>3</b>
<b>3.1. Features.....</b>	<b>3</b>
<b>3.2. Advantages.....</b>	<b>4</b>
<b>4. Servlet .....</b>	<b>4</b>
<b>4.1. Life cycle .....</b>	<b>4</b>
<b>4.2. Servlets advantages versus CGI .....</b>	<b>5</b>
<b>Bibliography.....</b>	<b>6</b>

# 1. Tomcat

Apache Tomcat is a **servlet container** (execution shell that handles and invoke servlets) developed below **Jakarta project** in the **Apache Software Foundation**. Tomcat implements specifications from servlets and JSP (Java Server Pages).

Tomcat is developed by the **Apache Software Foundation** and independent volunteers. It was first released in 1999 and his original founder was James Duncan Davidson.

His source code and his binaries are freely available for the users below the **Apache Software License**. First distributions were 3.0.x and the more recent versions are 9.x, implementing Servlet 4.0 and JSP 2.3.

## 1.1. Environment

Tomcat is a **web container** with support for servlets and JSPs. It includes the **Jasper compiler** that converts JSPs in servlets. Often is presented combined with the **Apache Web Server**.

Tomcat can work as web server itself and is user as autonomous web server in high-level traffic and high disponible. How is it developed in Java it can be used in any operating system that have the **Java Virtual Machine (JVM)**.

## 1.2. Directory structure

Directory	Description
/bin	Tomcat core. Mainly contains scripts for start and stop.
/classes	Used for append additional classes to the classpath.
/common	Common classes used by Catalina and the web applications.
/conf	Configurations files.
/lib	Contains <i>jar</i> libraries.
/logs	Stores daily logs.
/server	Classes only used by Catalina.
/shared	Shared classes for the web applications.
/src	Servlet API font files.
/webapps	Web applications.
/works	Temporary work files.

## 1.3. Features

- It's open source. Is not necessary spend money for use it.
- It's regularly updated to make it compatible with other software.
- Supports SSL and can be configured to secure sensitive data.
- It's cross-platform compatible.
- It's lightweight, allowing it for run smoothly on most systems.

## 2. Jakarta project

The **Jakarta Project**, founded in 1999 creates and maintains open source software for the **Java platform**. Operates as an umbrella project below the **Apache Software Foundation**, and all products created by this project are liberated below Apache License.

**Jakarta project** is retired in 2011 because did not have more subprojects. Main **Jakarta Project** contributions include tools, software libraries and frameworks like:

- **BCEL**. A Java byte code manipulation library.
- **Cactus**. Server-side unitary testing framework for Java classes.
- **ECS**. Java API for element generating for different markup languages.
- **JCS**. A Java developed cache distribution system.
- **Slide**. Content repository used by WebDAV.

Some projects that were part of Jakarta but now are independent projects in the **Apache Software Foundation** are:

- **Ant**. An application building tool.
- **Maven**. A project building and generation tool.
- **Struts**. A web application development framework.
- **Tomcat**. A JSP/Servlet container.
- **Turbine**. A fast application development framework.

## 3. JSP (JavaServer Pages)

JavaServer Pages (JSP) is a server-side multiplatform technology that allows to create dynamic web content easily. JSP is like PHP, but uses Java language. To deploy and use JSP is required a servlet compatible web server such as Apache Tomcat.

JSP performance is similar to any equivalent servlet, because is compiled as a Java class. Java Virtual Machine (JVM) also compiles the required parts of code to machine code, making it more efficient than web technologies that executes purely interpreted code.

### 3.1. Features

Main JSP features are:

- Not only can be used HTML, also can be used XML or WML.
- Is easier to reuse components that realize more specific tasks with JavaBeans.
- Support dynamic content that reflect real-world conditions.
- Web applications can be created easily.
- It have capacity for instance any Java class.

## 3.2. Advantages

Main advantage of JSP is that Java is a general-purpose language that exceeds web scope and is apt for create classes that access data in a diffuse way, permitting to separate web applications in layers, leaving HTML generation to a JSP file.

Other JSP advantage is that inherits Java portability, making possible to execute applications in multiple platforms.

JSP files are executed in a JVM, allowing them to be used in any computer if it has a JVM installed. Each JSP is executed independently, persisting from a petition to other, so not waste time invoking it. This also allows it to realize some things in a more efficient way, e.g. database connection.

Really, JSPs are an alternative way to create servlets, because their code is traduced to Java servlet code first time that are invoked, returning HTML code that compounds requested web page.

JSP can be a high-level abstraction of Java servlets. JavaServer Pages are real-time translated to servlets, which are saved in cache and reused up to original JSP is modified.

## 4. Servlet

A **servlet** is a **Java module** used for extend capabilities of a server. It is like an **applet**, but executed in a server. Servlets can respond to any request type, but commonly are used for extend server-hosted web applications.

Most common servlets use is to generate dynamic web pages from petition parameters sent by web navigator.

We can develop a simple servlet that creates a simple web page greeting, from create another that connects to a database, encrypting information for his send.

A **Servlets Container** is the component that manages servlets and controls their life cycle. This is an additional component what is needed to install it for support this **Java application** type in **J2EE** environments. The most popular container is **Apache Tomcat**.

### 4.1. Life cycle

1. **Initialize the servlet.** When a server loads a servlet, executes servlet *init* method. Initialization process must be completed before manage client petitions and destroy the servlet.

Server calls *init* method once in servlet instance creation, and is not called again unless are reloaded the servlet. Servlet cannot be reloaded if is not destroyed with *destroy* method.

2. **Interact with clients.** After initialization, servlet can serve client petitions. These petitions are attended by the same servlet instance, so it have to be careful when accessing shared variables to skip sync problems between simultaneous requirements.

3. **Destroy the servlet.** Servlets are executed up to are destroyed by server o with system administrator petition. When a servlet is destroyed, is executed his *destroy* method. This method is executed once and can be called with requests in process, so

you have to wait them. Server not executes the servlet again if is not reloaded and initialized again.

## 4.2. Servlets advantages versus CGI

- They are more efficient and uses less resources. **CGI** uses a new process by petition. **Servlets** have executed a copy and creates a thread by petition, reducing memory consumption and request time.
- They have persistence, so that still active once is finished the petition.

## Bibliography

- Ejemplo sencillo de Servlet - ChuWiki.* (n.d.). Retrieved from [http://chuwiki.chuidiang.org/index.php?title=Ejemplo\\_sencillo\\_de\\_Servlet](http://chuwiki.chuidiang.org/index.php?title=Ejemplo_sencillo_de_Servlet)
- Introducción a los Servlets | Manual Web.* (n.d.). Retrieved from <http://www.manualweb.net/javaee/introduccion-servlets/>
- Jakarta Project - Wikipedia, la enciclopedia libre.* (n.d.). Retrieved from [https://es.wikipedia.org/wiki/Jakarta\\_Project](https://es.wikipedia.org/wiki/Jakarta_Project)
- Java Servlet - Wikipedia, la enciclopedia libre.* (n.d.). Retrieved from [https://es.wikipedia.org/wiki/Java\\_Servlet](https://es.wikipedia.org/wiki/Java_Servlet)
- JavaServer Pages - Wikipedia, la enciclopedia libre.* (n.d.). Retrieved from [https://es.wikipedia.org/wiki/JavaServer\\_Pages](https://es.wikipedia.org/wiki/JavaServer_Pages)
- Lenguajes de programación: JSP.* (n.d.). Retrieved from <http://aplicaciones-web-lenguajes-programaci.blogspot.com/2011/12/jsp.html>
- Servicio para la Gestión de Guías de Práctica Clínica en XML.* (n.d.). Retrieved from <http://bibing.us.es/proyectos/abreproy/11774/fichero/2.7.-+Tomcat.pdf>
- The Jakarta Site - The Apache Jakarta™ Project.* (n.d.). Retrieved from <https://jakarta.apache.org/>
- Tomcat - Wikipedia, la enciclopedia libre.* (n.d.). Retrieved from <https://es.wikipedia.org/wiki/Tomcat>
- What is Apache Tomcat? | Comprehensive Guide to Apache Tomcat.* (n.d.). Retrieved from <https://www.educba.com/what-is-apache-tomcat/>