Servlets

Understanding Servlets

- A Servlet is a platform-independent Java program that runs within a server environment and interacts with users via a web browser.
- It is a server-side technology that processes requests and generates dynamic responses for web applications.
- Servlets are a crucial part of Java EE (Java Enterprise Edition) used for building dynamic web applications.
- Servlets are managed by a Servlet Container (also known as a Web Container), which handles the life cycle of servlets, including their initialization, request handling, and destruction.

Servlet Request-Response Workflow

Client Request : The user sends a request via a web browser (HTTP request)

Servlet Processing: Servlet processes the request (for example, querying a database).

Response Generation: Servlet generates an HTTP response and sends it back to the client.

Web Application:

- A Web Application is a software application that runs in a web environment and is accessed through web browsers.
- Web applications are commonly built using technologies like HTML, CSS, JavaScript, and Java Servlets.
- Web applications are deployed on a Web Container or Servlet Container within a server, which is responsible for managing servlets, handling requests, and generating responses.

Server:

- A server is the third party software developed by third party vendors according to SUN microsystems (Oracle) specification.
- It accepts requests from clients and provides appropriate responses..

- Servers are Classified in to ‘2’ types

  1. Web Server
  2. Application Server

Web Server:

A **Web Server** is a server designed specifically to host and execute Web Applications. It handles **HTTP** requests and responses. These servers have only a ***Web Container***, which accept request from HTTP protocol.

Example : Apache Tomcat

Application Server:

An **Application Server** is a server more comprehensive than a Web server. It is designed to handle both Web Applications and Enterprise Applications. They have BOTH a ***Web Container*** and an ***EJB Container***

Example : WebSphere, JBoss, IIS, WebLogic

Web Server Vs Application Server

Feature	Web Server	Application Server
Purpose	To host and serve web applications	To host web applications and enterprise-level applications
Containers	Only Web Container	Both Web Container and EJB Container
Protocols Supported	Mainly HTTP/HTTPS	HTTP, RMI, RPC, etc.
Complexity	Lightweight, suitable for basic web hosting	More complex, used for enterprise apps