

# **Java Web Development**

## **Servlets, JSP, and Java Web Frameworks**

*Laxman Nath*

*September 2024*

## 7.1 Servlets

### Web Container

- **Definition:** A web container (or servlet container) is a part of a web server that interacts with servlets. It manages their lifecycle and ensures they respond to web requests.
- **Functions:**
  - **Lifecycle Management:** Manages the initialization, execution, and destruction of servlets.
  - **Request Handling:** Handles client requests and sends responses back.
  - **Resource Management:** Provides access to resources such as databases, files, etc.

### Introduction to Servlets

- **Servlets:** Java classes that handle HTTP requests and generate responses. They are server-side components used for web development.
- **Uses:**
  - Processing user requests from web forms.
  - Managing sessions and maintaining state across multiple requests.
  - Interacting with databases and other backend resources.

### Servlet Lifecycle

1. **Loading and Instantiation:**
  - The servlet is loaded by the web container and an instance is created.
2. **Initialization (`init` method):**
  - The `init()` method is called once when the servlet is first loaded to perform any required initialization.
3. **Request Handling (`service` method):**
  - The `service()` method is called for each client request. It delegates the request to the appropriate `doGet()`, `doPost()`, etc., based on the request type.

## The Servlet APIs

- **Key Interfaces:**
  - `Servlet`, `ServletRequest`, `ServletResponse`
  - `HttpServletRequest`, `HttpServletResponse`
- **Common Methods:**
  - `doGet()`, `doPost()`
  - `getParameter()`, `getAttribute()`, `setAttribute()`

## Writing Servlet Programs

- **Steps:**
  - Implement `javax.servlet.Servlet`
  - Override `doGet` or `doPost`
  - Use `HttpServletRequest` and `HttpServletResponse`
- **Example:** Basic Servlet handling GET and POST requests.

## Handling Forms and HTTP Requests

- **Reading Form Parameters:**
  - `request.getParameter("paramName")`
- **Processing Forms:**
  - Use `doGet()` or `doPost()` methods.
- **Handling GET and POST:**
  - Differences in request handling.
  - Example code snippets.

## Database Access with Servlets

- **Steps:**
  - Load JDBC driver.
  - Establish connection with `DriverManager.getConnection()`.
  - Execute queries with `Statement` or `PreparedStatement`.
  - Handle results with `ResultSet`.
- **Example:** Simple database operation using a servlet.

## Handling Cookies and Sessions

- **Cookies:**
  - Create, set, and retrieve cookies.
  - `Cookie` class and methods.
- **Sessions:**
  - Managing sessions with `HttpSession`.
  - `session.setAttribute()`, `session.getAttribute()`.
- **Example:** Implementing user login with cookies and sessions.

## 7.2 JSP (JavaServer Pages)

### Servlets vs JSP

- **\*\*Servlets:\*\***
  - Controller logic.
  - Complex business logic handling.
- **\*\*JSP:\*\***
  - View layer.
  - Simplified for HTML content generation.
- **\*\*Comparison:\*\***
  - Use case differences.
  - Integration in MVC architecture.

### JSP Syntax and Basics

- **\*\*JSP Syntax:\*\***
  - **\*\*Directives:\*\*** `<%@ directive %>`
  - **\*\*Declarations:\*\*** `<%! declaration %>`
  - **\*\*Expressions:\*\*** `<%= expression %>`
  - **\*\*Scriptlets:\*\*** `<% code %>`
  - **\*\*Comments:\*\*** `<%-- comment --%>`
- **\*\*JSP Implicit Objects:\*\***
  - ``request`, `response`, `session`, `application`.`
- **\*\*Object Scope:\*\***
  - ``page`, `request`, `session`, `application`.`