

Document Generated: 12/15/2025

Learning Style: Virtual Classroom

Technology: Java

Difficulty: Intermediate

Course Duration: 5 Days

## Mastering Jakarta EE Web Development (TT5100)



## About this course:

Jakarta EE Web Application Development is a five-day hands-on course geared for experienced Java developers new to JEE, who need to get up and running with essential dynamic web development skills. Created in collaboration with several leading JEE / Java EE authors and industry experts, this comprehensive course teaches students how to design and program web components, including all the important concepts and hands on labs that will have you building working server-side applications in no time flat. This course provides core Jakarta EE / JEE knowledge and skills that can be used as the foundation for developing production-quality web applications to a basic level.

Servlets are a key server-side Java technology for building web applications. Servlets are programs that run on a web server; they can respond to client requests and create dynamic content. Servlets allow flexible generation of dynamic content. Additional technologies allow one to separate static from dynamic content while harnessing the power of servlets. Enhancements in JEE simplify web application development, supporting the use of annotations, context dependency injection (CDI), and a Common Expression Language (for use with both JSPs and JSF).

In 2017 Oracle gave the open source edition of the Java Enterprise edition to the Eclipse Foundation. Since the names 'Java' and 'javax' are still owned by Oracle, the open source version of Java Enterprise Edition (Java EE) has been renamed Jakarta EE. In addition, individual specifications were also renamed. JavaServer Faces (JSF) was renamed to Jakarta Faces (or simple 'Faces'), Java Enterprise Beans was renamed to Jakarta Enterprise Beans and so on.

## Course Objective:

- Design and build web applications from both business and technical requirements
- Build web interfaces with Jakarta Faces, JSPs and Servlets, using the latest technologies in JEE.
- Write maintainable web applications that separate HTML and Java
- Understand the design and development of web applications using Servlets, JSPs, web fragments, and JSF
- Use dependency injection (CDI) within their application
- Make Servlets cooperate and share data
- Store and process session information
- Deal with concurrency issues
- Access databases using an Object to Relational (ORM) framework

- Work with annotations included in Jakarta EE
- Work with WebSockets as well as asynchronous servlets
- Use Jakarta Bean validation in a web application
- Properly handle various types of exceptions

## **Audience:**

- This is an introductory- level Java programming course, designed for experienced developers who wish to get up and running with Java EE / JEE, or who need to reinforce sound Java for Web coding practices.

## **Prerequisite:**

- This is an introductory- level Java programming course, designed for experienced developers who wish to get up and running with Java EE / JEE, or who need to reinforce sound Java for Web coding practices.

## **Course Outline:**

Session: Developing Enterprise Applications

Lesson: Enterprise Development

- Enterprise Application Software
- Requirements of Enterprise applications
- Scalability, Load Balancing, Fail Over
- Resource pooling

Lesson: Jakarta EE Core Components

- Overview of Jakarta EE Core Components
- Web Tier Components
- Application Tier
- Deployable Units
- Deployment Descriptors
- The Java Naming and Directory Interface (JNDI)
- Tutorial: Building Web Applications in Eclipse

Session: JEE Dynamic Web Applications

Lesson: Introduction to Servlets

- The Servlet Interface
- The Web Container
- Creating HTML Output Using Servlets
- The @WebServlet Annotation
- Interaction Between web.xml and Annotations
- The @WebInitParam Annotation
- Lab: A First Servlet

#### Lesson: Form processing using Servlets

- Using HTML5 Forms with Servlets
- Processing Request Parameters
- HttpServletRequest Methods
- HttpServletResponse Methods
- Lab: Form Processing

#### Lesson: Jakarta Server Pages

- Jakarta Server Pages ( JSPs)
- The Relationship Between JSPs and Servlets
- The JSP lifecycle
- Lab: A First JSP

#### Lesson: Implementing MVC in JEE

- Model View Control
- Using the RequestDispatcher
- Handling Requests
- The Request Scope
- Handling Request Attributes
- The Expression Language (JSR 341)
- EL in Template text
- Lab: Implementing MVC

#### Lesson: Session Management

- Sessions in Web Applications
- The HttpSession object
- Session Management in JEE
- Handling Cookies
- URL-Rewriting
- Lab: Managing Sessions

#### Session: JEE Servlet Filters and Listeners

##### Lesson: Servlet Filters

- Introduce Servlet Filters
- Modify the request data
- Modify the response data

- The @WebFilter annotation
- Define Filter Mappings
- Move functionality out into a decorator pattern
- Lab: Adding Filters

#### Lesson: Events, Listeners and Initializers

- Introduce Web Listeners
- Listen for context events
- Respond to Session modifications
- Session aware objects
- ServletContextInitializer and the Service Provider interface
- The HandlesTypes annotation
- Lab: Listeners

#### Session: Jakarta Expression Language (EL)

##### Lesson: Overview of EL

- The Expression Language (JSR 341)
- Value and Method Expressions
- Immediate and Deferred Evaluation Syntax
- Read and Read/Write expressions

##### Lesson: The EL language

- Apply EL operators
- Use the EL implicit objects
- Explain the steps involved in creating EL functions
- Create a function implementation
- Define the function in the tag-library
- Use a function within a JSP page
- Lab: Using EL Expressions

#### Session: Custom Tags

##### Lesson: Introduction to Custom Tags

- Custom tags
- Using the taglib Page Directive
- The TLD File
- The Tag Implementation Class

##### Lesson: Jakarta Standard Tag Library

- Use the JSTL core and formatting libraries
- The core functionality of the library
- JSTL functions
- Lab: Using JSTL

## Session: Contexts and Dependency Injection (CDI)

### Lesson: Introduction to CDI

- Understand the value of CDI
- Explore dependency injection (DI)
- Understand alternatives
- Understand annotation processing
- Use and configure CDI
- Lab: Using CDI

### Lesson: Using CDI

- Use qualifiers to discriminate which object gets injected
- Understand when a bean is assignable to a given injection point
- Define your own Qualifier annotation
- Understand post construction annotations and pre destruction annotations
- Create factory methods with @Produces
- Lab: Using Qualifiers

### Lesson: CDI and Jakarta EE

- CDI's Relationship to Jakarta EE
- The @Model annotation
- Built-in CDI scopes
- Lab: Using CDI and Servlets

## Session: Using Resources

### Lesson: JEE DataSources

- DataSources in JEE
- Setup a DataSource
- Using CDI to inject a DataSource
- Lab: Using DataSources

### Lesson: Overview of JPA

- Object to Relational (O/R) Mapping (ORM)
- The Jakarta Persistence Architecture
- The ORM framework configuration
- Map a 'simple' entity to a database table
- Read, write and search for entities
- Lab: Using JPA

## Session: Java API for WebSocket

### Lesson: Introduction to WebSocket

- Java API for WebSocket Overview

- Using WebSockets in JEE
- Endpoint Instances

#### Lesson: Implementing WebSocket Endpoint

- Annotated Endpoints
- Receiving messages
- Send Response to Client(s)
- JavaScript to Setup a WebSocket Connection
- Lab: Implementing a WebSocket

#### Lesson: Extending WebSockets

- Understand the use of configurators
- Share user data between multiple requests
- Use JSON message objects for communication
- Write message encoders and decoders
- Handle exceptions in endpoints
- Lab: Encoding and Decoding Messages

#### Session: Jakarta Bean Validation

##### Lesson: Introduction to Bean Validation

- Bean Validation
- Define Constraints on Object Models
- Core Validation Annotations
- Validate Objects and Object Graphs
- Internationalized error messages
- Lab: Bean Validation

##### Lesson: Bean Validation

- Define custom validation constraints
- Implement constraint validators
- Use validation groups
- Use bean validation on methods and constructors
- Lab: Creating Constraints

#### Session: Managing Web Applications

##### Lesson: Web Fragments

- Need for Web Fragments
- The web-fragment Element
- Fragment Ordering
- Lab: Fragments

##### Lesson: Error Handling

- Handle exceptions in web applications
- Declaring error pages

#### Lesson: Asynchronous Servlets

- Invoking a 'Long Running' Process
- The `asyncSupported` Attribute
- Using the `AsyncContext` Class
- Handling `AsyncEvent` Objects
- Nonblocking I/O in Servlets
- Lab: ASync Servlets

#### Lesson: Web Security

- Specify the Servlet Security Model
- Roles in the Web Application
- Access Control and Authentication Requirements
- Security-Related Annotations
- Lab: Web Security

#### Session: Introduction to Jakarta Faces

##### Lesson: Introduction to Faces

- Faces Overview
- The Faces 'Components'
- Configuring a Faces Application
- MVC using Jakarta Faces
- Lab: First JSF

##### Lesson: JSF Components

- Understand the component architecture of JSF
- Explain the use of the `RenderKit`
- User Interface Component Model
- Introduce the JSF Custom Tags
- Explain the functionality of the various input tags
- Panels and tables in JSF
- Lab: JSF HTML Tags

#### Session: Facelets

##### Lesson: Facelets

- Introduce Facelets
- Use Facelets to create the view of the JSF application
- Access properties of a Managed Bean using EL
- Lab: Working With Facelets

##### Lesson: Facelets Templating and Resources



- Creating a Consistent Look and Feel
- Templating and Placeholders
- JSF resource management
- Lab: Facelets Templating