

Full Stack Java Developer Journey

Mastering Jakarta EE Web Development

Gain the core Jakarta EE knowledge and skills that are the foundation for developing professional-grade web applications

Course Snapshot

- **Course: Mastering Jakarta EE Web Development (TT5100)**
- **Duration:** 5 days
- **Audience & Skill Level:** The content is appropriate for experienced Java developers.
- **Hands-on:** This course is approximately 50% hands-on lab to lecture ratio. Student machines are required.
- **Delivery Options:** This course is available for **in-person presentation, live online / virtual presentation**, or can be presented in a **blended learning or short course format**.
- **Public Schedule:** This course has active dates on our open enrollment **Public Schedule**.
- **Customizable:** This course agenda, topics and labs can be further adjusted to target your specific training skills objectives, tools and learning goals. Please ask for details.

Overview

Unlock the power of dynamic web development with our five-day, hands-on **Mastering Jakarta EE Web Application Development** course. Perfect for experienced Java developers new to JEE, this course will take you on a journey of discovery as you learn the essential skills and best practices needed to build robust and scalable web applications. This comprehensive course is packed with engaging concepts, practical lab exercises, and real-world examples that will have you creating your own server-side applications in no time.

Jakarta EE is a Java-based platform for enterprise web development, which includes a set of specifications and technologies for building and deploying web applications. It provides a standard set of APIs and tools for developing server-side components such as servlets, JavaServer Faces (JSF), Enterprise JavaBeans (EJB), and Java Persistence API (JPA). Jakarta EE can be used in full stack web development to build the back-end or server-side of a web application, which can then be integrated with front-end technologies such as HTML, CSS, and JavaScript to create a complete web application. Some of the key benefits of using Jakarta EE for full stack web development include its scalability, robustness, and support for different deployment environments.

Why is JEE / Java EE now called Jakarta EE? 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.

Learning Objectives

You'll will leave this course armed with the required skills to design and build scalable, secure, maintainable web applications - leveraging our extensive experience in the delivery of scalable enterprise applications with complex web interfaces based on modern Java EE technologies.

Throughout the course, you'll gain the skills required to design and build scalable, secure, and maintainable web applications. You'll learn how to solve common web design problems, use JEE design patterns, and work with a variety of JEE and web technologies such as Servlets, JSPs, JSF, JNDI, and CDI. You'll also explore servlet capabilities, advantages, architecture, and session management, as well as how to manage resources, deploy, and build robust web applications using servlets and other components. The course starts with web application architecture and covers various web components used to create dynamic applications. Learn not just specific topics and APIs, but also how to integrate them into a complete application.

Working within in an engaging, hands-on learning environment, guided by our expert team, attendees will learn to:

- 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

Need different skills or topics? If your team requires different topics or tools, additional skills or custom approach, this course may be further adjusted to accommodate. See our **Java Full Stack Developer Journey** courses for additional topics and skills. Our team will collaborate with you to adjust the content to focus on your specific learning objectives and goals.

Audience & Pre-Requisites

The course is geared for experienced Java developers.

Take Before: Students should have practical skills equivalent to or should have attended the following course(s) as a pre-requisite:

- TT2104: Fast Track to Core Java Programming for OO Developers (C+, C#, etc.)

Next Steps / Follow On Courses: We offer a wide variety of follow-on courses for next-level Java development skills, Java for Web / Full Stack, Java EE, Spring, REST, Microservices, Unit Testing / TDD, Java secure coding, mobile development and more. Please see our **Java Developer Journey Courses & Learning Paths** for options based on your specific role and goals.

Enhanced Learning Services: Please also ask about our **Pre-Training Class OnRamp & Prep / Primer** offerings, **Skills Gap Assessment Services, Case Studies, Knowledge Check Quizzes, Skills Immersion Programs & Camps, Collaborative Mentoring Services** and **Extended Learning Support & Post Training** services.

Course Topics / Agenda

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 you 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

Lesson: Interceptors and Decorators

- Introducing CDI Interceptors
- Implementing Interceptors
- Adding (Multiple) Decorators
- CDI Events
- Lab: Using Interceptors and Decorators

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

Student Materials & Lab Environment

All course software (limited versions, for course use only), digital courseware files or course notes, labs / data sets and solutions (as applicable) are provided for you in our "easy access / no install required" high-speed remote lab environment. Our tech team works with every student to ensure everyone is set up with working access and ready to go prior to every course start date, ensuring a smooth delivery and great hands-on experience. Please ask for details.

For More Information

For more information about our dedicated training services, collaborative coaching services, courseware licensing options, public course schedule, training management services, partner programs, or to see our complete list of course offerings and special offers please visit us at www.triveratech.com, email Info@triveratech.com or call us toll free at **844-475-4559**. Our pricing and services are always satisfaction guaranteed.

TRIVERA TECHNOLOGIES • Collaborative IT Training, Coaching & Skills Development Solutions
www.triveratech.com • toll free +1-844-475-4559 • Info@triveratech.com • Twitter TriveraTech

ONSITE, ONLINE & BLENDED TRAINING SOLUTIONS • PUBLIC / OPEN ENROLLMENT COURSES • COURSEWARE LICENSING & DEVELOPMENT MENTORING • ASSESSMENTS • LEARNING PLAN DEVELOPMENT • SKILLS IMMERSION PROGRAMS / RESKILLING / NEW HIRE / BOOT CAMPS PARTNER & RESELLER PROGRAMS • CORPORATE TRAINING MANAGEMENT • VENDOR MANAGEMENT SERVICES

Trivera Technologies is a Woman-Owned Small-Business Firm

Ask About our new **Learning Journeys & Learning Experience Platform (LXP)**

Project-Based Training for Job-Ready Skills

