

Specialized - Mastering JEE Web Application Development

Code:	TT5100
Length:	5 days
URL:	View Online

Mastering JEE Web Application Development is a five-day hands-on JEE / Java EE training 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 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).

Skills Gained

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 JSF, 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
- Work JEE's version of dependency injection (CDI)
- Make Servlets cooperate and share data
- Store and process session information
- Deal with concurrency issues
- Access databases with JPA
- Work with annotations included in JEE
- Work with WebSockets as well as asynchronous servlets
- Use Java Bean validation in a web application
- Properly handle various types of exceptions

Who Can Benefit

This is an introductory- level Java programming course, designed for experienced developers who wish to get up and running with JEE, or who need to reinforce sound Java for Web / JEE coding practices. Attendees should have a working knowledge of developing basic Java software applications.

Prerequisites

Attendees should have incoming skills equivalent to those in the course(s) below or should have attended these as a prerequisite:

- TT2104 Fast Track to Java Programming for OO Experienced Developers

Course Details

Session: Developing Java EE applications

Lesson: Enterprise Development

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

Lesson: Java EE Core Components

- Overview of Java EE Core Components
- Web Tier Components
- Application Tier
- Deployable Units
- Deployment Descriptors
- The Java Naming and Directory Interface (JNDI)
- Tutorial: Working with Eclipse (JEE Version) and Apache TomEE 7.0.x
- Tutorial: Working with TomEE 7

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: Java Server Pages

- Java Server Pages (JSPs)
- The Relationship Between JSPs and Servlets
- The JSP lifecycle
- The role of JSPs in Java EE 7
- 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 Java EE
- 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

Session: Expression Language 3.0 (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

- EL Operators
- EL Reserved Words
- EL Implicit objects
- Referencing Objects using EL
- Lab: Working with EL

Session: Custom Tags

Lesson: Introduction to Custom Tags

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

Lesson: The Java Standard Tag Library

- JSTL Overview
- Use JSTL to Iterate Over Collections of Data
- JSTL functions
- Lab: Using JSTL

Session: Contexts and Dependency Injection (CDI)

Lesson: Introduction to CDI

- Context Dependency Injection (CDI)
- The @Inject Annotation
- The @Default Annotation
- The @Alternative Annotation
- The @Named Annotation
- Lab: Using CDI

Lesson: Using CDI

- Qualifiers
- @PostConstruct and @PreDestroy
- The @Produces Annotation
- Lab: Using Qualifiers

Lesson: CDI and Java EE

- CDI's Relationship to Java 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

- Introduce the Java Persistence API (JPA)
- Benefits of Using an ORM framework
- Hibernate and JPA
- Lab: Using JPA

Session: Java API for WebSocket

Lesson: Introduction to WebSocket

- Java API for WebSocket Overview
- Using WebSocket in Java EE
- Endpoint Instance

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

- Manage Client State
- Encoding and Decoding Messages
- Handling Errors
- Lab: Encoding and Decoding Messages

Session: Java Bean Validation (JSR 349)

Lesson: Introduction to Bean Validation

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

Lesson: Bean Validation

- Validate Parameters and Return Values
- Develop Custom Constraints
- 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

- Handling HTTP Errors Codes
- Handling Exceptions

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
- Servlet 3.1 Predefined Roles
- The deny-uncovered-http-methods XML Element
- Lab: Web Security

Session: Introduction to Java Server Faces

Lesson: Introduction to JSF

- JSF Overview
- The JSF 'Components'
- Configuring a JSF Application
- MVC using JSF
- 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

- Facelets as View Handlers
- Custom Tags Used in Facelets
- The @Named Annotation
- The @RequestScoped and @SessionScoped Annotations
- Using the dataTable Custom Tag
- Lab: Working With Facelets

Lesson: Facelets Templating and Resources

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