

Building Java / JEE Web Applications with Servlets, JDBC, JNDI and More (TT5100)

Modality: Virtual Classroom

Duration: 5 Days

About this Course:

This intermediate-level 5 Days Training Program is designed for Experienced Java Developers & Professionals seeking to master Web Application Development in Java Enterprise Edition. This course helps professionals nurture the essential web development skills and provides them with first-hand experience of working with Context Dependency Injections, Annotations, and Expression Languages. On average, a Java Developer earns \$90,992 annually.

This course is designed to train professionals with the essential skills needed to program & design web components and manage Web Server Servlets. Professionals will also learn the art of creating & managing dynamic content and effectively distinguish between static and dynamic content. The primary objective of this course is to help professionals build familiarity with every key web development feature & functionality in Java Web Application Architecture.

Course Objectives:

The core objective of this course is to help professionals develop a better understanding and sound knowledge of the following key concepts:

- Fundamentals and Essentials of Java EE Application Model & Java Servlets
- Introduction to Java Server Pages, JSTL, JSF & Expression Language (EL)
- Contexts and Dependency Injections (CDI)
- Introduction to Java Bean Validation, Web Socket API, & Java Persistent API
- Planning, Designing, & Development of a Web Application
- Web Interface Development with JSPs, Servlets, & JSF using Latest JEE Technologies
- Servlet Collaboration and Data Storage, Sharing, & Session Information Processing
- Concurrency Problem Resolution and JPA Database Access
- Getting to Know Java Enterprise Edition Annotation & WebSockets
- Working with Asynchronous Servlets & Java Bean Validation

Audience:

This course is specifically tailored for the following group of professionals and interested candidates:

- Seasoned Java Developers
- Web Developers & App Developers
- Professional striving to build familiarity with Java Enterprise Edition (JEE)
- Candidates seeking to enhance knowledge of Java Web Coding Practices

Prerequisites:

Professionals planning to enroll in the Building Java / JEE Web Applications with Servlets, JDBC, JNDI and More (TT5100) course should have fundamental knowledge and understanding of creating Java Software Apps

Course Outline:

Module 1: 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 Oxygen (JEE Version) and Apache TomEE 7.0.x
- Tutorial: Working with TomEE 7

Module 2: 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
- Exercise: A First Servlet

Lesson: Form processing using Servlets

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

- Exercise: 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
- Exercise: 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
- Exercise: Implementing MVC

Lesson: Session Management

- Sessions in Web Applications
- The HttpSession object
- Session Management in Java EE
- Handling Cookies
- URL-Rewriting
- Exercise: Managing Sessions

Module 3: 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
- Exercise: Adding Filters

Module 4: 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
- Exercise: Working with EL

Module 5: 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
- Exercise: Using JSTL

Module 6: 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
- Exercise: Using CDI

Lesson: Using CDI

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

Lesson: CDI and Java EE

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

Module 7: Using Resources

Lesson: JEE DataSources

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

Lesson: Overview of JPA

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

Module 8: Java API for WebSocket

Lesson: Introduction to WebSocket

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

Lesson: Implementing WebSocket Endpoint

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

Lesson: Extending WebSockets

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

Module 9: 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
- Exercise: Bean Validation

Lesson: Bean Validation

- Validate Parameters and Return Values
- Develop Custom Constraints
- Exercise: Creating Constraints

Module 10: Managing Web Applications

Lesson: Web Fragments

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

Module 11: Introduction to Java Server Faces

Lesson: Introduction to JSF

- JSF Overview
- The JSF 'Components'
- Configuring a JSF Application
- MVC using JSF
- Exercise: 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
- Exercise: JSF HTML Tags

Module 12: 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
- Exercise: Working With Facelets

Lesson: Facelets Templating and Resources

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