

Java Enterprise Edition: Integrating JSF, EJB and JPA - 4 Days Streamlining Enterprise Application Development

Course 936 Overview

- You Will Learn How To**
- Architect, develop, deploy and secure enterprise applications with Java EE
 - Utilize the JavaServer Faces (JSF) event-driven architecture to build rich user interfaces
 - Manage transactions and application security using annotations and XML
 - Effectively build business logic using Enterprise JavaBeans (EJB)
 - Integrate the Java Message Service (JMS) into applications with Message Driven Beans (MDB)
 - Achieve high-performance data persistence with Java Persistence API (JPA)

Course Benefits Java Enterprise Edition offers a revolutionary, simplified approach to enterprise application development. In this course, you gain in-depth knowledge of the major components of this technology: JSF, EJB and JPA. Through intensive hands-on exercises, you learn to implement high-performing, scalable and secure transactional applications.

Who Should Attend Those involved in developing enterprise applications with Java. Knowledge of Java at the level of Course 471, "Java Programming Comprehensive Introduction," is assumed. Knowledge of Web technologies and HTML is helpful.

Hands-On Training Throughout this course, extensive hands-on exercises based on an evolving case study provide you with practical experience in developing applications with Java EE. Exercises include:

- Building rich Web-based interfaces using JSF components
- Creating scalable objects using Session Beans
- Storing and retrieving data objects with JPA
- Extracting persistent data using JPQL
- Building asynchronous message receivers with MDB
- Managing complex transaction and security requirements with annotations

Java Enterprise Edition: Integrating JSF, EJB and JPA - 4 Days

Streamlining Enterprise Application Development

Course 936 Outline

Enterprise Application Development

- Establishing the components of an enterprise architecture
- Streamlining the development process with Java EE

Leveraging JavaServer Faces (JSF)

JSF architecture

- Identifying the JSF core components
- Exploring the request processing cycle
- Managing application flow using the JSF navigation model

Constructing a JSF application

- Creating views with JSF custom tags
- Handling user events with backing beans

Enhancing functionality using JSF services

- Applying server-side data validation
- Taking advantage of the Unified Expression Language

Generating Rich User Interfaces (UI) with JSF Component Technology

Building views with facelets

- Defining the functionality of the HTML component set
- Arranging the UI component layout
- Localizing messages using Resource Bundles

Modularizing UI development

- Simplifying page structure with Facelets
- Achieving code reuse with templates
- Exploring the composite component architecture

Leveraging Ajax to improve the user experience

- Sharpening response with the asynchronous Web model
- Combining Ajax functionality with JSF
- Utilizing prebuilt Ajax-enabled JSF components

Implementing the BusinessTier with EJB

Stateless and stateful beans

- Encapsulating scalable business logic with JavaBeans
- Accessing session beans remotely
- Constructing effective stateful services

Applying advanced strategies of session beans

- Adding behaviors with method interceptors
- Linking services with annotation-based resource injection
- Triggering timer-based and asynchronous services

Unleashing Message Driven Beans (MDB)

- Simplifying robust message receivers with MDB
- Generalizing message reception with Java connectors

Opening Access with Web Services

Achieving interoperability with JAX-WS

- Coding and packaging a service endpoint
- Packaging and deploying the service

Accessing services with JAX-WS clients

- Analyzing the client contract
- Binding and exchanging valid data types

Mapping with Java Persistence API (JPA)

Demystifying the JPA architecture

- Identifying the major components of JPA
- Establishing access with the EntityManager

Applying JPA core operations

- Developing the persistence class
- Storing Java objects
- Holding conversational state across requests

Handling Complex Object Relationships Effectively representing object associations

- Capturing single and multivalued associations
- Representing Java collections

Choosing appropriate inheritance strategies

- Employing techniques for class-to-database mapping
- Auditing operations with Callbacks and Listeners

Working with JPQL

- Initiating data access with the Java Persistence Query Language (JPQL) and API
- Selecting Entity and relationship properties

Querying with the Criteria API

- Creating a criteria query and managing results
- Enhancing queries with CriteriaBuilder

Enhancing application structure and performance

- Improving structure with named queries
- Augmenting JPQL with native optimized SQL

Preserving Integrity with Enterprise Services

- Approaches to working with Java Transaction API (JTA)
- Managing transaction strategies with annotations
- Securing each tier in a multitier architecture
- Configuring roles and access control