

Java EE 7: Back-End Server Application Development

Duration: 5 Days

What you will learn

The Java EE 7: Back-End Server Application Development training teaches you how to build and deploy enterprise applications that comply with Java Platform, Enterprise Edition 7 Full Profile. Learn to develop applications with the following technologies: Enterprise JavaBeans (EJB), Java Persistence API (JPA), JDBC, Java Transaction API (JTA), Contexts and Dependency Injection (CDI), Java Message Service (JMS), Bean Validation, Batch API, Timer services, Java EE Concurrency and more.

Learn To:

Use Java EE 7 technologies to create, read, update and delete database records using both JDBC and JPA technologies.

Create a flexible component model using EJB and CDI technology.

Create SOAP-based and XML web services.

Develop the business and integration tiers of an enterprise application.

Understand how those components responsible for: interacting with other systems through web services and message queues.

Become proficient with database access and manipulation using transactions.

Provide timer, concurrency and batch services.

Develop expertise using Java Enterprise Edition 7, the latest version of the Java platform for development of enterprise applications.

Benefits to You

When you walk away from this course, you will have developed the knowledge and skills to read and write messages to systems that may or may not be developed using Java with Java Message Service create batch services to process thousands of jobs in parallel. This interactive, hands-on training is an excellent follow-up course to the Java EE 7: Front-end Application Development training.

Related Training

Required Prerequisites

Understand OO principles

Basic understanding of database concepts and SQL syntax

Experience with Java SE

Java SE 8 Programming

Suggested Prerequisites

Java EE 7: Front-end Web Application Development

Java SE 7 or 8 programmer certification

Course Objectives

Apply dependency injection using CDI

Apply the batch API to the problem of processing thousands of jobs in parallel

Create and apply Timer services

Create and use web services in enterprise applications

Develop enterprise components using EJB

Use JDBC in an enterprise environment

Use JMS to communicate between various enterprise systems

Use JPA to persist entities and create

read

update and delete database records

Course Topics

Java Platform, Enterprise Edition

The Java EE Platform

The needs of enterprise application developers

Java EE specifications
A comparison of services and libraries
Java EE application tiers and architecture

Enterprise Development Tools and Applications

The purpose of an application server
Properties of Java EE components
The development process of Java EE applications
Configuring and deploying Java EE applications

Java Beans, Annotations and Logging

Java SE features in Java EE applications
Creating POJO JavaBeans components
Using logging
Using common Java annotations
Developing custom annotations
The role of annotations in Java EE applications

XML Programming with JAXB

The benefits of XML
XML namespaces and schemas
Java XML APIs
The Java XML Binding API (JAXB)
Reading and writing XML documents with JAXB
xjc: the JAXB binding compiler
JAXB annotations

SOAP Web Services with JAX-WS

Overview of SOAP
Overview of WSDL files
Comparing WSDL-first and code-first design approaches
Writing a JAX-WS web service
Generating WSDL from a Java class
Creating JAX-WS web service clients

Java Naming and Directory (JNDI) Services

What is JNDI?
Naming service concepts
Directory service concepts
JNDI packages
Using JNDI to look up JDBC and EJB components in Java EE

The EJB Component Model

The role EJB components play in Java EE applications
The role of the EJB container
EJB changes in Java EE 7
Local, distributed and no-client EJB client access views
EJB Session types
Stateless, Stateful and Singleton EJBs
Session bean packaging and deploying

Contexts and Dependency Injection

What is dependency injection?

Using Qualifiers

The beans.xml file and Alternatives

Using Producers and Disposers

Using Interceptors

Using Events and Stereotypes

Java Message Service

What is the Java Message Service?

Why do we need JMS?

JMS Overview

Point-to-point messaging architecture

Publish/subscribe messaging architecture

Message producers and consumers

Queues and topics

Durable vs. non-durable subscriptions

Message-driven Beans

The life cycle of a message-driven bean

Creating a message-driven bean

Creating life cycle handlers for message-driven beans

Configuring a message-driven bean

Java EE Concurrency

Concurrency in Java EE

Asynchronous EJBs

Managed Executors

JDBC in Java EE Environments

Overview of the JDBC API

Using CDI to inject a JDBC resource in a Java EE component

The Data Access Object pattern

Transactions in Java EE Environments

What are transaction semantics?

Comparing programmatic and declarative transaction scoping

Using JTA to scope transactions programmatically

Implementing a container-managed transaction policy using declarations

Controlling container-managed transaction propagation

Java Persistence API

Object-relational mapping

Entities and the entity manager

Persistence contexts and persistence units

Create, read, update and delete operations with JPA

Create typed queries in JPA with JPQL

Bean Validation with JPA

What is Bean Validation?

JPA lifecycle phases where validation takes place

Using the built-in validation constraints

Creating a custom bean validation constraint

Programmatic validation by injecting a Validator
Using validation groups

Timer and Batch Services

What are timer services?

Programmatic and automatic timers

What is Batch processing?

Jobs, steps and chunks

Batch examples

Security

Authentication, authorization and confidentiality

Apply Java EE security using deployment descriptors

Creating users and groups and mapping them to roles

Defining possible web service attack vectors