

Updated on 12/17/2024

Sign up

Java EE Training

5 days (35 hours)

Presentation

Java EE or Java Enterprise Edition is a server-side platform built on the Java language and the Java SE platform. This platform enables you to create, deploy and run web applications more easily on an application server.

The large number of libraries is one of the many [advantages of this platform](#). It's an advanced technology, or an advanced version of Java, specifically designed for the development of web-based, network-centric or enterprise applications.

It includes concepts such as Servlet, JSP, JDBC and Socket programming. Most applications developed with Java EE use a two-tier architecture (client and server).

Our Java EE training course will strengthen your skills in writing, deploying and testing Java EE applications. You'll master all Java EE specifications to develop robust, professional enterprise applications.

As with all our training courses, this one will introduce you to the latest version of Java 23 ([JDK 23](#)).

Objectives

- Know the basic principles of Java EE
- Develop applications on the Java EE platform
- Learn about and implement the Java EE platform architecture
- Mastering templates and deployment automation
- Use all the features of servlets, EJBs and JSPs
- Create components for each specification
- Mastering JEE technologies
- Control memory resources and processes using APIs
- Implement a data access layer and a presentation layer

Target audience

- Developers
- Technical architects
- Applications Development Manager
- IT Project Manager

Prerequisites

- Ideally, you should have taken our [Java 22 training course](#).
- Basic knowledge of HTML and CSS

Java EE training program: Developing enterprise applications

Java EE fundamentals

- What is Java EE?
- Discover the development environment
- Platform specifications
- Create your first Java application
- Servlets and JSPs
 - How do I create a Servlet?
 - JSP inclusion
 - Communicate data between your pages
 - Expression Language in JSP
- Setting up JSTL
 - Variables
 - Conditions
 - Loops
- Using databases with JDBC
- Processing data with SQL

Introduction to Java EE architecture

- First layer: Servlets and controllers
 - Communication between the application and external services
 - Securing
- Second layer: Business code
 - Business code services
 - Application code
 - Interaction with any caches
- Last layer: Data access
 - DAO
 - Files
 - External services

- Packaging and deployment of Java EE modules and applications
- Java EE application server operating principles

Deployments

- Web, EJB and client containers
- Composition of a JEE application deployment
- Simplify development
- Deployment models
- Deploying your application
- Use of data sources
- Configuring JavaMail providers and sessions
- Automate deployments with Docker and Ansible

Java EE specifications

- EJB (Enterprise Java Beans)
- JPA (Java Persistence API)
- JMS (Java Messaging Service)
- REST services with JAX-RS
- CDI (Contexts and Dependency Injection)
- Securing applications with JAAS

EJB: Enterprise Java Beans

- Java EE basic architecture
- Implementing Java EE Web services
- Main EJB components
 - How do I create a component?
 - Developing components
 - Exposing EJB components
- Different types of EJB
 - The sessions
 - Message-Driven
- Developing message-managed beans
- XML RPC invocation to components
- JBoss application integration with .NET

Messaging with JMS

- Create messaging applications
- Create mail clients
- Send and receive messages with the JMS API

JPA: Java Persistence API

- Creating persistent entities with validation
 - Entity life cycle
 - Entity mapping
 - Entity management with EntityManager
- Query language
- Lazy fetch optimization
- Annotations
- XML overloads

REST services

- Creating a REST service
- Developing REST services in Java
- Identify resources with URI
- Choice of data format (HTML, XML, JSON)
- Developing a resource class
- JAX-RS API
- Annotations

Securing enterprise applications

- Securing Java applications
- How do I use JAAS for a Java EE application?
- System lock
- Implementing HTTPS

JSF Framework

- Introducing JSF
- Operating principles
- General file settings
- Implementing ManagedBeans
- Facelets
- JSF Booksellers
- Conversions and validations
- Internationalization

Struts Framework

- Introduction to the Struts framework
- General parameters
- Basic operation
- Project organization
 - First pages
 - Actions
- URL settings
- Graphics components (OGNL)
- Implementing Struts

- Adding dependencies

Socket programming

- Socket interface
- TCP programming (Linux)
 - Exchange diagram
 - Server connection
 - Create client-side socket
 - Data exchange
- UDP programming (Linux)
 - Exchange diagram
 - Local socket attachment
 - Creating a UDP server
- Socket TCP programming (Windows)
 - Socket creation
 - Server connection
 - Data exchange
 - Building a TCP server

Tomcat application server

- What is Tomcat?
- Download and installation
- Deploying an application on Tomcat

Monitoring with JMX (Java Management Extensions)

- Introducing JMX
- Architectural overview
- Configuring Java extensions
- Local monitoring
- Authentication management
- Enhancing safety
- Troubleshooting

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as enrolment is finalized, the learner receives a self-assessment questionnaire enabling us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and objectives.

This questionnaire also enables us to anticipate any connection or internal security problems (intra-company or virtual classroom) that could be problematic for the follow-up and smooth running of the training session. This questionnaire also enables us to anticipate any connection or internal security difficulties (intra-company or virtual classroom) that could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.

