

Java for Web Development

Intermediate Java, Spring Boot and REST APIs

FA-0633 · 4 days

Software Development

Apply existing Java skills to a guided Spring Boot web application with REST endpoints, persistence and practical error handling.

Introduction

This four-day course is for active Java programmers who understand object-oriented programming and can build and debug a Java project. It reviews data handling and concurrency before introducing Spring, Spring Boot and a small REST application.

Participants extend one prepared project with layered request handling and Spring Data JPA using a disposable H2 database. The course introduces local configuration, embedded servers and selected development/health tools; it does not promise advanced JVM expertise or production-ready deployment.

Learning outcomes

The course teaches participants to:

- Apply control flow, bounded recursion and data conversion to small Java tasks.
- Work with collections, generics, streams and basic concurrent tasks.
- Handle errors and resources in file and threaded examples.
- Explain dependency injection and the flow through a small layered Spring application.
- Create a Spring Boot project using compatible Java and build-tool versions.
- Build and exercise GET, POST, PUT and DELETE endpoints with a simple persistence layer.
- Review configuration, development tools, health information and limitations before any production use.

Prerequisites

- Recent Java programming experience, including classes, objects, compiling and debugging.

- Ability to use a Java IDE and familiarity with files and basic server concepts.
- A prepared compatible JDK, Maven, IDE and project dependencies; internet access for authorised dependency downloads.
- Ability to use SSH if the selected exercise uses a supplied remote training server; a local training environment is also suitable.

Training outline

1. Day 1 — Java Data-Handling Review

- Revision of OOP
- Loops
- Conditions
- Recursion, termination and practical limits.
- Data conversion, including a bounded nested JSON or XML example.
- Data Manipulation
- Validate and clean a small prepared dataset.

Review object/reference lifetime and automatic memory management as needed for the examples.

2. Day 2 — Collections, Concurrency and Web Project Preparation

Collections and Concurrent Code

- Collections
- Compare lists and arrays, mutability and access patterns.
- Generics
- Functional interfaces and lambdas.
- Stream processing; parallel execution is not automatically faster.
- Threads
- Multi-threaded programming
- Synchronisation and shared-state risks.

Errors, Files and Task Utilities

- Exception Handling
- Error Handling

- Exception propagation and failures in a small multithreaded example.
- Java file APIs, resource handling and filesystem-specific behaviour.
- Executor/task utilities appropriate to the selected JDK.

Project and Web Overview

Review the prepared project, exercise selected file and database operations and introduce the request/response flow of the small web application.

3. Day 3 – Spring and Spring Boot

Spring Project Structure

- Intro with Spring Framework
- Define a small training project and its boundaries.
- Create one compatible Spring/Spring Boot project with Spring Initializr.
- Build the prepared project incrementally.
- Coupling and separation of responsibilities in a small example.
- Overview of background processing and task boundaries.
- Terminology
- Trace a request through controller, service and persistence layers.
- Maven dependency and build workflow.
- Dependency injection with emphasis on understandable component wiring.

Spring Boot Application

- Getting Started with Spring Boot
- Inspect the Boot project structure rather than creating an unrelated second project.
- Hello World API with Spring Boot
- Request/response data conversion and validation in the prepared example.
- Configuration
- Embedded Servers
- Spring Boot Actuator: inspect local health information and discuss controlled exposure of management information.
- Spring Boot DevTools for local development, not a production requirement.

4. Day 4 — Persistence and REST Endpoints

Spring Data JPA and H2

- Simple REST API with Spring Boot and Spring Data JPA
- Jakarta Persistence entities and a disposable H2 in-memory database; do not treat the training database as production persistence.
- Additional Configuration for Spring Data JPA
- SQL
- Entity and integrating with GET REST API

REST Operations

- Understanding REST API
- Building REST API (GET)
- Building REST API (POST)
- Building REST API (PUT)
- Building REST API (DELETE)

Assessment

Exercise the prepared API, inspect stored sample data and explain selected design, configuration and error-handling choices.