Faysal Aziz

Java - Starting to Deployment

FA-0269 · 3-day

Software Development

Introduction

Build a solid Java core and power up to web apps in Spring Boot - all in three intensive days.

This 3-day course is designed for developers and aspiring professionals who want direct, industry-relevant training. The instructor brings over 30 years of experience and focuses on real-world patterns, frameworks and tasks—not textbook abstractions. We'll start by grounding you in the solid foundations of Java, then rapidly transition into building a web application using Spring Boot (the latest version) so you leave with both the core language skills *and* the framework capability.

Learning outcomes

By the end of this course you will be able to:

- Write clean, maintainable Java code using modern language features and object-oriented principles.
- Use Java's key APIs (collections, streams, exceptions, I/O) in real-world scenarios.
- Understand the architecture of Spring Boot and set up a Spring Boot project from scratch.
- Build a web application using Spring Boot: define controllers, services, repositories, deploy with embedded server.
- Integrate persistence (JPA or Spring Data) and expose RESTful endpoints.
- Apply best practices for configuration, dependency injection, error handling and project structure in a Spring Boot web app.

Prerequisites

- Basic programming experience (any language) — variables, loops, functions/methods.
- Familiarity with using an IDE (e.g., IntelliJ IDEA, Eclipse) and command-line tools.
- Understanding of HTTP and basic web concepts (requests/responses) is helpful but not mandatory.

- A computer with JDK installed (Java 17 or later recommended) and an IDE of your choice.
- Approach: ready to code, ask questions, and apply hands-on examples.

Training outline

1. Introduction & Setup

1. Course overview, goals, and expectations
2. Environment setup: JDK version, IDE, build tools (Maven or Gradle)
3. Project structure, version control basics

2. Java Language Fundamentals

1. Data types, variables, operators
2. Control flow: if/else, switch, loops (for, while, do)
3. Methods and basic modular code

3. Object-Oriented Programming in Java

1. Classes and objects
2. Constructors, fields, methods
3. Access modifiers, encapsulation
4. Inheritance and polymorphism
5. Interfaces, abstract classes
6. Composition vs inheritance

4. Core Java APIs and Tools

1. Exception handling (try/catch/finally, throws, custom exceptions)
2. Collections framework: List, Set, Map – basic operations
3. Java Stream API and lambda expressions (modern Java)
4. I/O, Files, Paths (brief overview)
5. Basics of concurrency (threads, executors) – introduction only

5. Modern Java Features & Best Practices

1. Java versions and LTS releases (Java 17, 21, etc)

2. Local variable type inference (var), switch enhancements, records/sealed types (if applicable)
3. Code style, modularisation, tests (unit testing overview)
4. Debugging, logging, and profiling basics

6. Transition to Web Application Development

Introduction: Why use Spring Boot?

1. Framework overview: Spring Boot provides embedded servers, starter dependencies, auto-configuration.
2. Web applications, REST services and microservices context.

Setting up a Spring Boot project

1. Using Spring Initializr, choosing dependencies
2. Build tool configuration (Maven/Gradle)
3. Project skeleton, main application class annotated with @SpringBootApplication

7. Spring Boot Web Layer

1. Creating controllers (@RestController, @GetMapping, @PostMapping)
2. Request/response handling, path variables, query params
3. Exception handling in Spring Boot (global handler)
4. Packaging and running applications (embedded Tomcat/Jetty)
5. Configuration (application.properties / application.yml), environment profiles

8. Data Access and Persistence

1. Introduction to JPA and Spring Data repositories
2. Defining entities, relationships (one-to-many, many-to-one)
3. CRUD operations via Spring Data interfaces
4. Transaction management

9. Building a Complete Web Application

1. Designing service layer and repository layer
2. Layered architecture: controller ➜ service ➜ repository
3. API endpoints and data model mapping
4. Handling validation, DTOs vs entities
5. Simple UI (optional) or Postman testing of endpoints

10. Production-Readiness and Advanced Topics Overview

1. Logging, monitoring, and actuator endpoints (health, metrics)
2. Packaging and deployment options (jar, war, containerization)
3. Introduction to Spring Boot 3.x new features (native images, observability)
4. Security basics (authentication, authorization) – outline only
5. Next steps: microservices architecture, cloud deployme

© 2026 Faysal Aziz · Kuala Lumpur