Faysal Aziz

From Zero to Java 21 Microservices

Build Intelligent Systems in 3 Days

FA-0217 · 3 Days

Agentic & Generative AI · Software Development

Introduction

Learn Java from scratch, master Spring Boot microservices, Kafka messaging, and AI/ML integration - fast, modern, and industry-ready

---

The demand for Java developers remains strong, but today's roles require more than just knowing the syntax — modern systems are built as **microservices**, integrate with **streaming platforms like Kafka**, and increasingly depend on **AI/ML services** to personalize and automate experiences.

This 3-day immersive course takes complete beginners in Java and rapidly brings them up to speed with **Java 21**, the most modern and powerful release yet. We'll cover the essentials of programming in Java, then transition into **Spring Boot microservices** with real-world communication patterns. Finally, we'll explore how to integrate **Kafka** for event streaming and bring intelligence into systems with **AI APIs, agentic AI, and ML-based recommendation services**.

With over **30 years of industry experience**, your instructor ensures this course is not academic fluff — it is built around **what companies need right now**.

Learning outcomes

By the end of this course, participants will:

- Understand the fundamentals of Java programming (syntax, OOP, collections, exceptions, multithreading).
- Use **modern Java 21 features** (records, pattern matching, virtual threads) in practical coding tasks.
- Build and run a **Spring Boot microservice** with REST endpoints and database integration.

- Implement **asynchronous communication** between services using **Kafka**.
- Call **external AI APIs** from Java microservices to enrich business logic.
- Build a **basic ML-based recommendation service** and integrate it into the system.
- Understand how **agentic AI** can automate microservice operations and decision-making.
- Deliver a **mini-project** that ties all concepts together.

Prerequisites

- No prior knowledge of Java is required.
- General familiarity with computers and basic coding logic (e.g., what variables and loops are) will help.
- Laptop with **Java 21 JDK**, **IDE (IntelliJ / Eclipse / VS Code)**, **Maven/Gradle**, and **Docker** pre-installed (setup guide provided before the course).
- Curiosity and willingness to learn at an accelerated pace.

Training outline

1. 1. Foundations of Java 21 Programming

Getting Started

- Installing Java 21 and IDE setup
- Running the first Java program (Hello World)
- Understanding compilation and execution

Core Syntax & Basics

- Variables, data types, operators
- Control structures (if, switch, loops)
- Functions & methods
- Input/output in Java

Object-Oriented Programming (OOP)

- Classes and objects
- Constructors, methods, and fields
- Inheritance and polymorphism
- Encapsulation and abstraction
- Interfaces vs abstract classes

Core Java Libraries

- Collections framework (List, Set, Map)
- Sequenced Collections (Java 21)

- Exception handling (try, catch, finally)
- File handling and streams

Modern Java 21 Features

- Record classes for data objects
- Pattern matching for switch
- Virtual threads for concurrency (Project Loom)
- String templates and unnamed variables (preview features)

Hands-on Exercises

- Build a simple console-based banking system
- Use collections to store and retrieve user data
- Run concurrent tasks with virtual threads

---

2. 2. Introduction to Spring Boot & Microservices

Spring Boot Essentials

- What is Spring Boot and why microservices?
- Project structure, starters, and auto-configuration
- Creating RESTful APIs with Spring MVC
- Connecting to databases with Spring Data JPA
- Basic error handling and validation

Microservices Concepts

- Monolith vs microservices
- Service boundaries and responsibilities
- Communication patterns: synchronous vs asynchronous
- Deploying microservices with Docker

Hands-on Exercises

- Build a User Service microservice (CRUD API)
- Build an Order Service microservice connected to a database
- Containerize both services with Docker

---

3. 3. Kafka and Event-Driven Microservices

Kafka Basics

- Topics, partitions, producers, consumers
- Kafka vs REST communication

- Event-driven system design

Spring Boot + Kafka Integration

- Configuring Kafka in Spring Boot
- Producing and consuming messages
- Handling errors and retries
- Event logging and monitoring

Hands-on Exercises

- Publish order events from Order Service
- Consume order events in Inventory Service
- Implement an asynchronous notification service

---

4. 4. AI, ML & Agentic AI Integration

Calling AI APIs

- Overview of AI APIs (LLMs, NLP, computer vision)
- Making REST calls to AI APIs from Spring Boot
- Handling responses, caching, and errors

ML-Powered Recommendation Service

- Basics of recommendation systems (content-based, collaborative filtering)
- Simple implementation (pre-trained ML model or lightweight algorithm)
- Exposing recommendations via REST API
- Integrating recommendations into user flow

Agentic AI for Microservices

- What are agentic AI systems?
- Use cases in monitoring, automation, and decision-making
- Building a simple AI agent microservice to monitor logs and trigger alerts

Hands-on Exercises

- Build a Recommendation Service microservice
- Integrate with User Service to suggest products
- Add AI agent to monitor Kafka event failures and auto-restart services

---

5. 5. Capstone Project

© 2026 Faysal Aziz · Kuala Lumpur

https://faysalaziz.com/courses/from-zero-to-java-21-microservices/