Faysal Aziz

Advanced Java with Microservices and Design Patterns

Robust Java design, Kafka integration and service implementation

FA-0755 · 5 days

Software Development

Introduction

Develop stronger Java designs and apply them to event-driven integration and microservices. The course is intended for experienced Java developers and combines selected implementation exercises with demonstrations of specialist runtime APIs.

Across five days, work through behavioural, creational and architectural patterns, robust language techniques, Kafka clients and service communication. The breadth is a structured introduction to advanced practice, not a promise of complete mastery or production readiness.

Learning outcomes

- Apply selected design patterns and explain their trade-offs.
- Improve class, interface, generic and exception-handling designs.
- Use recursion and resource-conscious implementation practices, and recognise concurrency and deserialisation risks.
- Implement a small Kafka producer/consumer workflow and distinguish it from Kafka Streams processing.
- Use domain-driven design concepts to define a small microservice boundary.
- Compare and practise REST, gRPC and event-based service communication.
- Review security, deployment, maintenance and common distributed-system failure considerations.

Prerequisites

- Good understanding of OOP.
- Ability to program in Java.
- Ability to use IDE for Java.
- Compile and Deploy software in Java.
- Understand the entire development lifecycle in Java.

Comfort with command-line tools and the operating-system file system.

A supported JDK, suitable IDE/build tools and access to a prepared isolated Kafka and microservice lab. Only required installation permissions and approved network endpoints are needed; unrestricted root access or access to all ports is not required.

Training outline

1. Day 1 – Design patterns

- Introducing Design Patterns

  Behavioral Design Patterns
  - Introducing behavioral design patterns
  - Chain of responsibility pattern
  - Command pattern
  - Iterator pattern

  Creational Design Patterns
  - Introduction
  - Abstract factory
  - Builder
  - Singleton

  Architectural Patterns
  - Introduction
  - Event-driven pattern
  - Microservices pattern
  - Model-view-controller pattern

2. Day 2 – Advanced Java implementation

- Recursion, instance construction, static initialisation and object methods.

- Class and interface contracts, generics, enums and annotations.
- Efficient methods, coding guidelines and exception handling.
- Concurrency correctness and resource-use trade-offs.
- Built-in serialization, deserialization risks and filtering.
- Selected demonstrations of reflection, dynamic-language integration, compiler API and Java agents; account for the selected JDK and dependency versions.

3. Day 3 — Event-driven Java integration

- Access the prepared Kafka environment and inspect topics and compatible clients.
- Create and control producers, use serializers and compare multiple producers.
- Consumer partition assignment and offset commits, including auto-commit limitations.
- Refactor the example using relevant patterns and distinguish basic clients from Kafka Streams.

4. Day 4 — Microservice design and implementation

- Services, SOA and microservice trade-offs.
- Development environment, deployment and maintenance concepts.
- Domain-driven design concepts and a selected service boundary.
- Implement a small microservice and introduce appropriate service-security controls.

5. Day 5 — Communication and assessment

- Inter-process communication through REST.
- A gRPC service definition, generated client/server contracts and a guided request example.
- Event-based communication using the prepared messaging environment.
- Common problems and practical design decisions: failures, coupling, resource use and maintainability.
- Assess the implementation and document trade-offs and next steps.

© 2026 Faysal Aziz · Kuala Lumpur