

Training: Building Microservices

Duration: 5 Days

Audience: Java Developers with experience with developing monolith applications

Day 1: Introduction to Microservices & First Service

Module 1: Introduction to Microservices

- Microservices vs Monolithic Architecture
- Core principles of microservices
- Real-world use cases

Module 2: Your First Microservice

- Setting up the Java development environment (JDK, Maven/Gradle)
- Creating a basic Spring Boot application using Spring Initializr
- Solution and project structure for microservices (multi-module Maven/Gradle setup)

Module 3: Adding Database Storage

- Introduction to relational databases
- Setting up MySQL or PostgreSQL (or H2 for local dev)
- Using Spring Data JPA / Hibernate for data access

Lab: Build and run a microservice with database integration (e.g., Product Service) using Spring Boot + Spring Data JPA.

Day 2: Expanding the System & Communication Basics

Module 4: Preparing for the Next Microservice

- Creating additional microservices (e.g., Order Service)
- Independent deployments and service isolation

Module 5: Synchronous Inter-Service Communication

- REST API integration using RestTemplate / WebClient (Spring WebFlux) / OpenFeign
- Handling timeouts and retries (Resilience4j)

Module 6: Asynchronous Inter-Service Communication

- Basics of messaging systems (RabbitMQ or Apache Kafka) with Spring AMQP / Spring for Kafka
- Implementing event-based communication

Lab: Implement inter-service communication using both REST (Feign/RestTemplate) and asynchronous messaging (RabbitMQ/Kafka).

Day 3: Frontend and Microservice Security Introduction

Module 7: Initial Frontend Integration

- Building a simple frontend using HTML/JavaScript or Thymeleaf
- Calling microservices APIs from frontend

Module 8: Identity in Microservices

- Basics of authentication and identity providers (Keycloak)
- Understanding tokens (JWT) and claims

Module 9: Microservices Security Patterns and Techniques

- Gateway authentication (Spring Cloud Gateway)
- Service-to-service token validation
- Secure-by-default practices

Lab: Connect a frontend to microservices with secure token-based authentication.

Day 4: Security Implementation and Authorization

Module 10: Implementing Microservices Security

- Enabling authentication using Spring Security
- Securing API endpoints using JWT tokens (Spring Security OAuth2 Resource Server)

Module 11: Authorization in Microservices

- Role-based and policy-based access control (@PreAuthorize, method security)
- Using claims for custom authorization logic

Module 12: Frontend Integration to Secure Microservices

- Managing login and tokens in frontend

- Handling authorization failures and session expiry

Lab: Secure multiple services and integrate authenticated access in the frontend.

Day 5: Transactions and Store Experience

Module 13: Transactions with Microservices

- Challenges with distributed transactions
- Eventual consistency and reliability concerns

Module 14: Preparing the Saga Participants

- Saga design pattern fundamentals
- Setting up participant services (Order, Payment, Inventory) using Spring Boot

Module 15: Implementing the Purchase Saga

- Orchestration-based implementation (Spring State Machine or custom orchestrator)
- Handling success/failure paths

Module 16: Compensation and Idempotency

- Undoing actions when part of a flow fails
- Ensuring operations are idempotent (idempotency keys, database constraints)

Module 17: The Frontend Store Experience

- Integrating complete purchase flow
- Displaying products, placing orders, handling responses

Final Lab: Full store experience including secure login, order placement, and recovery – built entirely on Spring Boot microservices.