

## Training program:

# Microservices in Java

### Info:

<b>Name:</b>	<b>Microservices in Java</b>
<b>Code:</b>	<b>Java-ms</b>
<b>Category:</b>	Java and JVM
<b>Target audience:</b>	tech_lead developers architects
<b>Duration:</b>	3 days
<b>Format:</b>	30% lecture / 70% workshop

---

The training introduces participants to microservices, discussing key concepts of dependency management and project generation. Participants learn how to create projects, manage configuration, and refresh the application configuration as it runs.

Participants will also focus on microservices communication. They will learn about service discovery, HTTP resilient communication and message-based communication. Practical exercises include creating HTTP applications, using circuit breakers and writing API gateways.

The training also focuses on application observability. Participants learn to monitor applications, create metrics, track distributed systems and test stability using chaos engineering. The training also covers the use of feature toggles.

### Trainees will learn to

- create and manage microservices
- ensure application reliability and scalability through techniques such as circuit breakers and message-based communication
- monitor applications using metrics and distributed tracking

### It's all about the content.

- training provided by a maintainer of microservices tools
- wide range of training topics - from configuration, to communication and observability
- current technologies and advanced techniques such as chaos engineering or feature toggles

# Training program

## 1. Day 1: Introduction to microservices

- 1.1. Introduction to microservices using the Spring Boot framework as an example
- 1.2. Refreshing configuration during the life of an application
- 1.3. Application configuration management
- 1.4. Microservices on the Kubernetes platform

## 2. Day 2: Microservices communication

- 2.1. Exploring services
- 2.2. Resilient HTTP communication
- 2.3. Message-based communication
- 2.4. Writing the Gateway API

## 3. Day 3: Application observability

- 3.1. Application metrics with Micrometer
- 3.2. Observability through abstraction
- 3.3. Chaos engineering
- 3.4. Using feature toggles