

# Course : Creating a Microservices Architecture in Java

Practical course - 3d - 21h00 - Ref. RUL

Price : 2150 CHF E.T.

 3,8 / 5

BEST

## Teaching objectives

At the end of the training, the participant will be able to:

- Defining a microservices-based architecture
- Developing and consuming REST services
- Setting up microservices using a Java framework
- Deploying microservices in Docker containers
- Administering and monitoring microservices

## Course schedule

### 1 Introduction

- A look at upgrades in software and organization.
- Pros and cons of monolithic applications.
- Agile methodologies and DevOps practices.
- Continuous integration and continuous deployment processes.

### 2 Microservices architectures

- Defining a microservices architecture.
- Characteristics of microservices.
- The emergence of microservices architectures.
- The main players.
- Several examples from the web's biggest companies.

## PARTICIPANTS

## PREREQUISITES

## TRAINER QUALIFICATIONS

The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

## ASSESSMENT TERMS

The trainer evaluates each participant's academic progress throughout the training using multiple choice, scenarios, hands-on work and more. Participants also complete a placement test before and after the course to measure the skills they've developed.

## TEACHING AIDS AND TECHNICAL RESOURCES

- The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.
- At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.
- A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

## TERMS AND DEADLINES

### 3 Communications in a microservices architecture

- Choosing the collaboration style: REST request/reply vs Publish-Subscribe Messaging.
- Basic principles and refreshers on HTTP.
- Implementing REST services with Java.
- The principle of HATEOAS.
- Documenting a REST service with Swagger.
- Testing a REST service.

#### Hands-on work

Defining a Rest API with Swagger and implementing services with Jersey.  
Testing a REST service with SoapUI.

Registration must be completed 24 hours before the start of the training.

#### ACCESSIBILITY FOR PEOPLE WITH DISABILITIES

Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at [psh-accueil@orsys.fr](mailto:psh-accueil@orsys.fr) to review your request and its feasibility.

### 4 Developing a microservice in Java

- Overview of "all-in-one" Java frameworks.
- Comparison of SpringBoot and Dropwizard.
- Developing an application with Dropwizard or SpringBoot.
- Packaging a production-ready application.

#### Hands-on work

Developing an application with Dropwizard or SpringBoot.

### 5 Executing a microservice using containers

- Overview of Docker.
- How containers work with Docker.
- Designing custom images.
- Configuring and starting containers.

#### Hands-on work

Creating a custom container with Docker.

### 6 Deploying a microservices architecture

- Deployment optimization techniques.
- Implementing a multi-container application with Docker Compose.
- Infrastructure solutions for microservices (Kubernetes, Mesos, Swarm).
- Cloud solutions (Cloud Foundry, Heroku).

#### Hands-on work

Implementing a multi-container architecture with Docker Compose.

### 7 Administration and monitoring

- Centralizing logs.
- Monitoring microservices.
- Metrics and Health Check.

#### Hands-on work

Setting up an application's metrics to visualize them.

## Dates and locations

### REMOTE CLASS

2026: 9 Mar., 22 June, 22 June, 7 Oct., 7 Oct.,

23 Nov., 23 Nov.