

# Microservices and Micro-Frontends

## Course 3475 – 32 Hours

### Overview

---

Microservice architecture is an architectural style that structures an application as a collection of services that are highly maintainable and testable, Loosely coupled, Independently deployable, Organized around business capabilities and owned by a small team Micro-frontends are the answer to today's increasingly complex web applications. Inspired by the microservices model, this approach lets organizations break interfaces into separate features managed by different teams of developers.

In this practical workshop, you will learn how to design, implement and deploy applications based on microservices and micro-frontends approach. We will use mainly JavaScript ( Node.js, React & Angular ) programming languages and many other tools to build a complete application throughout the workshop

### Who Should Attend

---

- Developers
- DEVOPS

### Prerequisites

---

- JavaScript – Advanced knowledge
- Typescript – Basic knowledge
- Node.js – Basic knowledge
- React - Basic knowledge
- Angular – Basic knowledge
- HTML
- CSS

### Course Contents

---

#### Microservices and Micro-Frontends introduction

- What Are Microservices?
- Microservices At a Glance
- Key Concepts of Microservices
- Advantages of Microservices
- Should I Use Microservices?

#### Microservices and Micro-Frontends introduction

- What Are Microservices?
- Microservices At a Glance
- Key Concepts of Microservices

- Advantages of Microservices
- Should I Use Microservices?
- Micro-frontends overview & advantages

### **Microservices Architecture & Common Patterns**

- Communication
  - HTTP
  - Message Queue ( RPC )
- Databases & Domains
- Discovery URLs
- Gateway

### **Micro-frontends Architecture & Concept**

- Applying Microservices concept in frontend
- Define Micro-frontend
- Domain driven design
- Bounded context definition
- Micro-frontend routing
- Micro-frontend communication

### **Hands on**

- Building Node.js microservice
  - Writing Node.js Api Gateway
  - Writing a Node.js Microservice & communicate with Rabbitmq
  - Communicate with 3rd party services
- Building Micro-frontend Application
  - Creating a routing module with JavaScript
  - Creating Reacts Applications
  - Communicate with the Microservices gateway

### **Testing & Deployment**

- From unit to e2e tests
  - Unit testing
  - Integration testing
  - E2e testing
- Dockerizing node.js application
- Orchestrations – docker-compose
- Deployment
- Building and deploying Micro-frontends