

.Net Microservices Development

Course Summary

Description

The microservices architectural style promotes the development of complex applications as a suite of small services based on business capabilities. This course will help you identify the appropriate service boundaries within your business.

Objectives

After taking this course, students will be able to:

- Get acquainted with Microsoft Azure Service Fabric
- Compare microservices with monolithic applications and SOA
- Learn Docker and Azure API management
- Define a service interface and implement APIs using ASP.NET Core 2.0
- Integrate services using a synchronous approach via RESTful APIs with ASP.NET Core 2.0
- Implement microservices security using Azure Active Directory, OpenID Connect, and OAuth 2.0
- Understand the operation and scaling of microservices in .NET Core 2.0
- Understand the key features of reactive microservices and implement them using reactive extensions

Topics

- An Introduction to Microservices
- Implementing Microservices
- Integration Techniques and Microservices
- Testing Microservices
- Deploying Microservices
- Securing Microservices
- Monitoring Microservices
- Scaling Microservices
- Introduction to Reactive Microservices
- Creating a Complete Microservice Solution

Prerequisites

Basic computer skills, internet access, basic analytic or programming skills

Duration

Three Days

.Net Microservices Development

Course Outline

- I. **An Introduction to Microservices**
 - A. Origin of microservices
 - B. Discussing microservices
 - C. Monolithic architecture
 - D. Understanding the microservice architecture
 - E. Why should we use microservices?
 - F. How does the microservice architecture work?
 - G. Advantages of microservices
 - H. SOA versus microservices
 - I. Prerequisites of the microservice architecture
 - J. Understanding the problems with the monolithic architectural style
 - K. Prerequisites for microservices
 - L. Identifying decomposition candidates within monolithic
 - M. Overview of Azure Service Fabric
- II. **Implementing Microservices**
 - A. Introduction
 - B. Size of microservices
 - C. What makes a good service?
 - D. DDD and its importance for microservices
 - E. The concept of seam
 - F. Communication between microservices
- III. **Integration Techniques and Microservices**
 - A. Communication between services
 - B. Integration patterns
- IV. **Testing Microservices**
 - A. How to test microservices
 - B. Testing strategies (testing approach)
 - C. Testing pyramid
 - D. Types of microservice tests
 - E. Tests in action
- V. **Deploying Microservices**
 - A. Monolithic application deployment challenges
 - B. Understanding the deployment terminology
- C. Prerequisites for successful microservice deployments
- D. Isolation requirements for microservice deployment
- E. Need for a new deployment paradigm
- F. Containers
- G. Introducing Docker
- VI. **Securing Microservices**
 - A. Security in monolithic applications
 - B. Security in microservices
- VII. **Monitoring Microservices**
 - A. Instrumentation and telemetry
 - B. The need for monitoring
 - C. Monitoring challenges
 - D. Monitoring strategies
 - E. Logging
 - F. Monitoring in Azure Cloud
 - G. Other microservice monitoring solutions
- VIII. **Scaling Microservices**
 - A. Scalability overview
 - B. Scaling infrastructure
 - C. Microservice scalability
 - D. Scaling the infrastructure
 - E. Scaling service design
- IX. **Introduction to Reactive Microservices**
 - A. Understanding reactive microservices
 - B. Let's make code reactive
 - C. Event communication
 - D. Managing data
 - E. The microservice ecosystem
 - F. Coding reactive microservices
- X. **Creating a Complete Microservice Solution**
 - A. Architectures before microservices
 - B. Monolith transitioning
 - C. Monitoring
 - D. Monitoring strategies
 - E. Reactive microservices