AZ 2009: Build distributed apps with .NET Aspire

Course Duration: 8 hours (1 day)

Course description
Learn how to build observable, production ready, distributed applications with .NET Aspire. This course walks learners through setting up .NET Aspire, editing applications that are already using .NET Aspire, and creating their own .NET Aspire application.

Learning objectives
After completing this course, students will be able to:

- Describe .NET Aspire.
- Create a .NET Aspire project.
- Use telemetry in a .NET Aspire project.
- Use databases in a .NET Aspire project.
- Improve performance with a cache in a .NET Aspire project.
- Send messages with RabbitMQ in a .NET Aspire project.

Audience profile
This course is designed for .NET Developers who want to build expertise with .NET Aspire and building containerized solutions.

Audience prerequisites
Experience building web applications using .NET and C#.

Module 1: Introduction to .NET Aspire

Learn about the .NET Aspire stack in .NET 8 and how you can use it to ease and accelerate the development of cloud-native web applications.

- Introduction
- What is .NET Aspire?
- Learn about .NET Aspire integrations
- Learn about .NET Aspire tools
- Exercise - Use the .NET Aspire eShop

Module 2: Create a .NET Aspire project

Learn how to create cloud-native applications from scratch or add orchestration to an existing app by using the .NET Aspire stack in .NET 8.

- Introduction
- Learn how to create a new .NET Aspire project
- Exercise - Create a new .NET Aspire project
- How to add orchestration to an existing .NET app
- Exercise - Integrate an existing ASP.NET Core web app

Module 3: Use telemetry in a .NET Aspire project

Learn how to use telemetry in a .NET Aspire project to log data and examine the behavior of a cloud-native application.

- Introduction

- Telemetry APIs in .NET
- Using OpenTelemetry in .NET Aspire
- Using the .NET Aspire dashboard
- Using a logger to add custom log messages
- Exercise - Use telemetry in a .NET Aspire project

Module 4: Use databases in a .NET Aspire project

Learn about the database systems that .NET Aspire can connect to using built-in integrations. Then see how to configure connections to, and store data in, relational and nonrelational databases.

- Introduction
- Store data in SQL-compliant databases
- Store data in NoSQL databases
- Exercise - Use database services to persist data from a .NET Aspire project
- Create tests with .NET Aspire

Module 5: Improve performance with a cache in a .NET Aspire project

Learn about using Redis caching to increase performance in a cloud-native application. You'll also discover how .NET Aspire makes it easy to access Redis caching services from your microservices.

- Introduction
- Caches and Redis
- Using a Redis distributed cache
- Using a Redis output cache
- Customize connection strings by using .NET Aspire manifests
- Exercise - Cache data in Redis

Module 6: Send messages with RabbitMQ in a .NET Aspire project

Learn how to use message queues on RabbitMQ exchange servers to decouple microservices and manage communications in a cloud-native application built with .NET Aspire.

- Introduction
- What is RabbitMQ?
- Sending messages with RabbitMQ
- Exercise - Send messages between microservices through RabbitMQ