

DEVELOPING MICROSOFT AZURE AND WEB SERVICES

Module 1: Overview of service and cloud technologies

This module provides an overview of service and cloud technologies using the Microsoft .NET Core and the Azure. The first lesson, Key Components of Distributed Applications, discusses characteristics that are common to distributed systems, regardless of the technologies they use. Lesson 2, Data and Data Access Technologies describes how data is used in distributed applications. Lesson 3, Service Technologies, discusses two of the most common protocols in distributed system and the .NET Core technologies used to develop services based on those protocols. Lesson 4, Cloud Computing, describes cloud computing and how it is implemented in Azure

Module 2: Querying and Manipulating Data Using Entity Framework

In this module, you will learn about the Entity Framework data model, and about how to create, read, update, and delete data. Entity Framework is a rich object-relational mapper, which provides a convenient and powerful application programming interface (API) to manipulate data. This module focuses on the Code First approach with Entity Framework.

Module 3: Creating and Consuming ASP.NET Core Web APIs

ASP.NET Core Web API provides a robust and modern framework for creating Hypertext Transfer Protocol (HTTP)-based services. In this module, you will be introduced to the HTTP-based services. You will learn how HTTP works and become familiar with HTTP messages, HTTP methods, status codes, and headers. You will also be introduced to the Representational State Transfer (REST) architectural style and hypermedia. You will learn how to create HTTP-based services by using ASP.NET Core Web API. You will also learn how to consume them from various clients. After Lesson 3, in the lab "Creating an ASP.NET Core Web APIs", you will create a web API and consume it from a client.

Module 4: Extending ASP.NET Core HTTP Services

ASP.NET Core Web API provides a complete solution for building HTTP services, but services often have various needs and dependencies. In many cases, you will need to extend or customize the way ASP.NET Core Web API executes your service. Handling needs such as applying error handling and logging integrate with other components of your application and supporting other standards that are available in the HTTP world. Understanding the way ASP.NET Core Web API works is important when you extend ASP.NET Core Web API. The division of responsibilities between components and the order of execution are important when intervening with the way ASP.NET Core Web API executes. Finally, with ASP.NET Core Web API, you can also extend the way you interact with other parts of your system. With the dependency resolver mechanism, you can control how instances of your service are created, giving you complete control on managing dependencies of the services.

Module 5: Hosting Services On-Premises and in Azure

In this module you will learn how to host your application on-premises and on Azure. You will also learn about Docker containers, and writing serverless applications with Azure functions.

Module 6: Deploying and Managing Services

In this module, you will learn about Web Deploy and how to deploy web applications by using Web Deploy in Visual Studio. You will also learn how to define continuous integration and continuous delivery pipelines and how to use Azure API Management and OpenAPI to provide robust, secure, and reliable APIs to your customers.

Module 7: Implementing Data Storage in Azure

This module explains how to store and access data stored in Azure Storage. It also explains how to configure storage access rights for storage containers and content.

Module 8: Diagnostics and Monitoring

This module explains how to monitor and log services, both on-premises and in Azure.

Module 9: Securing services on-premises and in Microsoft Azure

This module describes claim-based identity concepts and standards, and how to implement authentication and authorization by using Azure Active Directory to secure an ASP.NET Core Web API service.

Module 10: Scaling Services

This module explains how to create scalable services and applications and scale them automatically using Web Apps load balancers, Azure Application Gateway and Azure Traffic Manager.