

# MVS3-7 - MOC 20487 - DEVELOPING WINDOWS AZURE AND WEB SERVICES

Categoria: **Visual Studio**

## INFORMAZIONI SUL CORSO



Durata:  
5 Giorni



Categoria:  
Visual Studio



Qualifica Istruttore:  
Microsoft Certified  
Trainer



Dedicato a:  
Sviluppatore



Produttore:  
Microsoft

## OBIETTIVI

After completing this course, students will be able to:

- Describe the basic concepts of service development and data access strategies using the .NET platform.
- Describe the Microsoft Azure cloud platform and its compute, data, and application hosting offerings.
- Design and develop a data-centric application using Visual Studio 2017 and Entity Framework Core.
- Design, implement, and consume HTTP services using ASP.NET Core.
- Extend HTTP services using ASP.NET Core.
- Host services on-premises and in Microsoft Azure.
- Deploy services to both on-premises and cloud environments and manage the interface and policy for their services.
- Choose a data storage solution, cache, distribute, and synchronize data.
- Monitor, log, and troubleshoot services.
- Describe claim-based identity concepts and standards, and implement authentication and authorization with Azure Active Directory.
- Create scalable service applications.

## PREREQUISITI

Before attending this course, students must have:

- Experience with C# programming, and concepts such as lambda expressions, LINQ, and anonymous types
- Understanding the concepts of n-tier applications
- Experience with querying and manipulating data with ADO.NET

## CONTENUTI

### Module 1: Overview of service and cloud technologies

- Key Components of Distributed Applications
- Data and Data Access Technologies
- Service Technologies
- Cloud Computing
- Manipulating Data

### Lab : Exploring the Work Environment

- Creating an ASP.NET Core project
- Create a simple Entity Framework model
- Create a web API class
- Deploy the web application to Azure

## **Module 2: Querying and Manipulating Data Using Entity Framework**

- ADO.NET Overview
- Creating an Entity Data Model
- Querying Data

### **Lab : Creating a Data Access Layer using Entity Framework**

- Creating a data model
- Query the Database

### **Lab : Manipulating Data**

- Create repository methods
- Test the model using SQL Server and SQLite

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

- HTTP Services
- Creating an ASP.NET Core Web API
- Consuming ASP.NET Core Web APIs
- Handling HTTP Requests and Responses
- Automatically Generating HTTP Requests and Responses

### **Lab : Creating an ASP.NET Core Web API**

- Create a controller class
- Use the API from a browser
- Create a client

## **Module 4: Extending ASP.NET Core HTTP Services**

- The ASP.NET Core Request Pipeline
- Customizing Controllers and Actions
- Injecting Dependencies into Controllers

### **Lab : Customizing the ASP.NET Core Pipeline**

- Use Dependency Injection to Get a Repository Object
- Create a Cache Filter
- Create a Debugging Middleware

## **Module 5: Hosting Services On-Premises and in Azure**

- Hosting Services on-premises
- Hosting Services in Azure App Service
- Packaging Services in Containers
- Implementing Serverless Services

### **Lab : Host an ASP.NET Core service in a Windows Service**

- Creating a new ASP.NET Core Application
- Registering the Windows Service

### **Lab : Host an ASP.NET Core Web API in an Azure Web App**

- Create a Web App in the Azure portal
- Deploy an ASP.NET Core Web API to the Web App

### **Lab : Host an ASP.NET Core service in Azure Container Instances**

- Publish the service to a Docker container

- Host the service in Azure Container Instances

#### **Lab : Implement an Azure Function**

- Develop the service locally
- Deploy the service to Azure Functions

#### **Module 6: Deploying and Managing Services**

- Web Deployment with Visual Studio 2017
- Continuous Delivery with Visual Studio Team Services
- Deploying Applications to Staging and Production Environments
- Defining Service Interfaces with Azure API Management

#### **Lab : Deploying an ASP.NET Core web service on Linux**

- Publish the ASP.NET Core web service for Linux
- Configure Nginx as a reverse proxy

#### **Lab : Deploying to Staging and Production**

- Deploy the application to production
- Create a staging slot
- Swap the Environments

#### **Lab : Publishing a Web API with Azure API Management**

- Creating an Azure API Management instance
- Testing and managing the API

#### **Module 7: Implementing Data Storage in Azure**

- Choosing a Data Storage Mechanism
- Accessing Data in Azure Storage
- Working with Structured Data in Azure
- Geographically Distributing Data with Azure CDN
- Scaling with Out-of-Process Cache

#### **Lab : Storing Files in Azure Storage**

- Store publicly accessible files in Azure Blobs
- Generate and store private files in Azure Blobs

#### **Lab : Querying Graph Data with CosmosDB**

- Create the CosmosDB graph database
- Query the CosmosDB database

#### **Lab : Caching out-of-process with Azure Redis cache**

- Create the Azure Redis Cache service
- Access the cache service from code
- Test the application

#### **Module 8: Diagnostics and Monitoring**

- Logging in ASP.NET Core
- Diagnostic Tools
- Application Insights

#### **Lab : Monitoring ASP.NET Core with ETW and LTTng**

- Collect and view ETW events
- Collect and view LTTng events

#### **Lab : Monitoring Azure Web Apps with Application Insights**

- Add the Application Insights SDK
- Load test the web service
- Analyze the performance results

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

- Explaining Security Terminology
- Securing Services with ASP.NET Core Identity
- Securing Services with Azure Active Directory

### **Lab : Using ASP.NET Core Identity**

- Add ASP.NET Core Identity middleware
- Add authorization code
- Run a client application to test the server

### **Lab : Using Azure Active Directory with ASP.NET Core**

- Authenticate a client application using AAD B2C and MSAL.js

## **Module 10: Scaling Services**

- Introduction to Scalability
- Automatic Scaling
- Azure Application Gateway and Traffic Manager

### **Lab : Load Balancing Azure Web Apps**

- Prepare the application for load-balancing
- Test the load balancing with instance affinity
- Test the load balancing without affinity

### **Lab : Load Balancing with Azure Traffic Manager**

- Deploy an Azure Web App to multiple regions
- Create an Azure Traffic Manager profile

## **INFO**

**Materiale didattico:** Materiale didattico in formato digitale

**Costo materiale didattico:** incluso nel prezzo del corso a Calendario

**Natura del corso:** Operativo (previsti lab su PC)