ASP.NET Core MVC (4043)

Duration: 3 days (24 hours)

This course provides a practical hands-on introduction to developing Web applications using ASP.NET Core 6.0 MVC and C#. This Web development framework from Microsoft emphasizes separation of concerns in the architecture and testability of applications. This course covers development on Microsoft's new .NET Core platform. The course is current to Visual Studio 2022 and.NET 6, which is the latest version of .NET Core and is the foundation for future Microsoft .NET development.

This course covers the fundamentals of the Model-View-Controller design pattern and its implementation in ASP.NET Core MVC. This technology is compared with classical ASP.NET Web Forms. Visual Studio 2022 is used as a productive platform for creating MVC Web applications. After presenting the fundamentals of the technology with several examples, the main components of Model, Controller and View are covered in detail. The discussion of the Model incorporates Microsoft technologies for persisting data, including XML Serialization and ADO.NET with SQL Server 2019. The routing mechanism of ASP.NET MVC is covered. The course includes an introduction to ASP.NET Web API and deployment on the Azure cloud.

Numerous programming examples and exercises are provided. The student will receive a comprehensive set of notes and all the programming examples.

Benefits

- Gain a thorough understanding of the philosophy and architecture of Web applications using ASP.NET Core MVC
- Gain a practical understanding of.NET Core
- Acquire a working knowledge of Web application development using ASP.NET Core MVC and Visual Studio 2022
- Persist data with XML Serialization and ADO.NET with SQL Server 2019
- Create HTTP services using ASP.NET Core Web API
- Deploy ASP.NET Core MVC applications to the Windows Azure cloud

Outline

1. Introduction to ASP.NET Core MVC

- Review of ASP.NET Web Forms
- Advantages and Disadvantages of Web Forms
- Model-View-Controller Pattern
- ASP.NET Core 6.0
- Considerations in Using ASP.NET MVC
- Unit Testing

2. Getting Started with ASP.NET Core MVC

- ASP.NET Core MVC Testbed
- Using Visual Studio
- Configuring for ASP.NET Core MVC
- Rendering Views
- Razor View Engine
- Dynamic Output

3. NET MVC Architecture

- The Controller
- The View
- The Model
- Helper Methods for HTML
- Form Submission
- Model Binding
- Input Validation

4. The Model

- More Complex Models in MVC Programs
- Microsoft Technologies for Model Persistence
- Using XML Serialization
- NuGet Package Manager
- Using ADO.NET

5. The Controller

- Controller Base Class
- Actions
- Retrieving Data from a Request
- Action Results
- Action Attributes
- Serving Static Files
- Filters

6. The View

- View Responsibility
- Using ViewBag
- Using Dynamic Objects
- Html Helpers
- Validation Attributes

7. Routing

- Routing in ASP.NET Core MVC
- Properties of Routes
- Parameters in Routing
- Registering Routes
- Attribute Routing

8. NET Core Web API

- ASP.NET Core Web API
- Representational State Transfer
- REST and Web API
- HTTP Services Using Web API
- HTTP Testing Tools
- Using Postman
- HTTP Response Codes

- ASP.NET Web API Clients

9. NET Core and Azure

- What Is Windows Azure?
- A Windows Azure Testbed
- Deploying an Application to Azure
- Updating an Application on Azure

System Requirements

Required software is Visual Studio, which includes .NET 6.0. The free Visual Studio Community 2022 may be used. Visual Studio 2022 includes LocalDB, a lightweight version of SQL Server 2019 Express, which is used for one example in the course. The operating system should be Windows 10 or more recent. The free Postman tool is used. It can be downloaded from the Web. See the course Setup Guide for details.

A good minimal hardware profile for this course would have a 1.8 GHz or faster CPU, 4 GB of RAM, and at least 20 GB of free disk space for tools installation and courseware.

Required Prerequisites

The student should have a good working knowledge of C# and the .NET Framework.

Student would require Azure Free Subscription

Useful Prerequisites

Basic knowledge of HTML and some prior exposure to .NET Core would also be helpful. Useful prior Object Innovations courses are Object-Oriented Programming in C# or C# Essentials plus .NET Foundations.