Introduction to ASP.NET Core MVC

Duration: 4 Days

US Price: $1795

Delivery Options: Attend face-to-face in the classroom, remote-live or via on-demand training.

Description

This course is designed to provide an introduction to .NET Core for programmers who already know the C# language. The course focuses on core portions of the .NET Framework that are common across many application areas. It starts with an introduction to the architecture and key concepts of .NET. The course then discusses class libraries, packages, metapackages and frameworks. Coverage includes working with delegates and events, I/O and serialization, memory management, processes and threads as well as threading and an introduction to the Task Parallel Library (TPL).

This course also provides a practical hands-on introduction to developing Web applications using ASP.NET Core MVC 6 and C#. This Web development framework from Microsoft emphasizes separation of concerns in the architecture and testability of applications. This course covers the fundamentals of the Model-View-Controller design pattern and its implementation in ASP.NET Core MVC. Visual Studio 2017 with ASP.NET Core 2.0 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 2016. The routing mechanism of ASP.NET MVC is covered. The course includes an introduction to ASP.NET Web API.

Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Prerequisites

C# programming experience.

Course Overview
## .NET Fundamentals
- What is Microsoft .NET?
- Common Language Runtime
- Framework Class Library
- Language Interoperability
- Managed Code
- .NET Core and Cross-Platform Development

## Class Libraries
- Components in .NET
- Comparing Components in COM and .NET
- Creating Class Libraries
- Using Components in Client Programs by Obtaining References to Class Libraries
- Using References

## Packages and Frameworks
- Overview of NuGet Packages and the NuGet Gallery
- Explaining the Role of Packages, Metapackages and Frameworks in .NET
- How Packages and Metapackages are Used in .NET Core
- Using the Visual Studio Package Manager to Manage Packages in Solutions
- Installing Packages from the NuGet Gallery
- Creating and Using Your Own NuGet Packages
- Porting from .NET 4.6 to .NET Core

## I/O and Serialization
- Using .NET Framework Classes for Working with Directories and Files
- Explaining and Using Streams for Performing File I/O
- Explaining the Role of Serialization in Persisting and Transporting Objects
- Serialization Mechanisms Available in .NET Core Contrasted with Classical .NET
- Using XML Serialization in .NET Core Programs

## Delegates and Events
- Using Delegate Objects to Implement Callbacks
- Using the `Random` Class to Generate Random Test Data
- Using Aggregations of Delegate Objects
- Using Delegate Objects to Implement and Handle Event Notifications

## .NET Programming Model
- Garbage Collection in .NET
- Implementing the Finalize/Dispose Pattern
- Using the Process and Thread Model for .NET Applications
  - Process and Thread Isolation
- Using the `Process` Class to Manage Processes in .NET Core Applications
- Working with Command-Line Arguments

## .NET Threading
- Using the `Thread` Class to Implement Multithreading in .NET Applications
- Using the `Monitor` Class to Program Safe Concurrent Access to Shared Data

## Overview of ASP.NET MVC
- Advantages and Disadvantages of ASP.NET Web Forms
- Understanding the Model-View-Controller (MVC) Pattern
- Outlining the Parts of an ASP.NET
View
- How Dynamic Objects Can be Used in ASP.NET Core MVC
- Using HTML Helpers
  - String Helpers
  - Link-Building Helpers
  - Form Helpers
  - Validation and Templated Helpers
- Using Validation Attributes in the Model

Routes in ASP.NET Core MVC
- Understanding the Use of Parameters in Routing
- Understanding How to Register Routes and the Importance of Order
- Using Attribute Routing to Map Actions Directly to Route Templates by Means of Attributes

ASP.NET Core Web API
- Overview of the ASP.NET Web API
- Overview of RESTful Web Services
- Implementing HTTP Services Using the Web API with ASP.NET Core MVC
- Using Postman to Exercise HTTP Requests while Developing a Web API Service
- Implementing Web API clients

ASP.NET Core and Azure
- What Is Windows Azure?
- A Windows Azure Testbed
- Deploying an Application to Azure
- Updating an Application on Azure