www.logicalimagination.training
800.657.1494

ASP.NET Core MVC

Course #: VC-530 **Duration:** 3 days

Prerequisites

Students should have a good understanding of HTML and CSS and be experienced C# developers.

Details

This course provides a practical hands-on introduction to developing Web applications using ASP.NET Core MVC 6 and C#. The course is current to .NET Core 2.0, which is a major update of the original .NET Core 1.0. Working with .NET Core is now simpler, as project information is consolidated in project files without the need for source JSON files. Also, the .NET Core 2.0 API has been extended and is a much larger subset of classical .NET than in .NET Core 1.0.

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. 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.

Software Needed

Required software is Visual Studio 2017 or higher, which includes LocalDB, a lightweight version of SQL Server 2016 Express. The free Visual Studio Community 2017 or higher may be used. The operating system should be Windows 7SP1 or more recent. Several free tools are also used. These can all be downloaded from the Web.

Outline

ASP.NET Core MVC

- **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 MVC
  - .NET Core
  - Considerations in Using ASP.NET MVC
  - Unit Testing
- **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
- **.NET MVC Architecture**
  - The Controller in ASP.NET MVC
  - The View in ASP.NET MVC
  - The Model in ASP.NET MVC
  - Helper Methods for HTML
  - Form Submission
  - Model Binding

    - Input Validation
- **The Model**
    - More Complex Models in MVC Programs
    - Microsoft Technologies for Model Persistence
    - Using XML Serialization
    - NuGet Package Manager
    - Using ADO.NET
- **The Controller**
    - Controller Base Class
    - Actions
    - Retrieving Data from a Request
    - Action Results
    - Action Attributes
    - Serving Static Files
    - Filters
- **The View**
    - View Responsibility
    - Using ViewBag
    - Using Dynamic Objects
    - HTML Helpers
    - Validation Attributes
- **Routing**
    - Routing in ASP.NET Core MVC
    - Properties of Routes
    - Parameters in Routing
    - Registering Routes
    - Attribute Routing
- **.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
- **.NET Core and Azure**
    - What is Windows Azure?
    - A Windows Azure Testbed
    - Deploying an Application to Azure
    - Updating an Application on Azure