

Course duration

- 4 days

Course Benefits

- Learn the skills needed to design and build applications that follow the MVC design pattern.

Available Delivery Methods

Public Class

Public expert-led online training from the convenience of your home, office or anywhere with an internet connection. Guaranteed to run .

Private Class

Private classes are delivered for groups at your offices or a location of your choice.

Course Outline

1. Introduction
 1. What is .NET Core?
 2. .NET Core vs .NET Full Framework
 3. Overview of ASP.NET Core
2. .NET Core SDK
 1. Installation
 2. Version Management
 3. Command-Line Interface (CLI)
 4. Hello World Application
3. ASP.NET Core Application Architecture
 1. NuGet Packages and Metapackages
 2. Application Startup
 3. Hosting Environment
 4. Middleware and the Request Processing Pipeline
 5. Services and Dependency Injection
4. Application Configuration
 1. Configure Method
 2. ConfigureServices Method
 3. MVC Components

- 4. Configuration Providers and Sources
 - 5. Configuration API
 - 6. Options Pattern
- 5. Request Routing
 - 1. RESTful API
 - 2. Routing Middleware
 - 3. Route Templates
 - 4. Route Constraints
 - 5. MVC Middleware
 - 6. Attribute-Based Routing
- 6. Models
 - 1. Introduction
 - 2. Persistence Ignorance
 - 3. Object-Relational Mapping
 - 4. Entity Framework (EF) Core
- 7. Controllers
 - 1. Introduction
 - 2. Requirements and Conventions
 - 3. Dependencies
 - 4. Action Results
- 8. Views
 - 1. Introduction
 - 2. Conventions
 - 3. Razor Syntax
 - 4. Layouts
 - 5. ViewData and ViewBag
 - 6. Strongly-Typed Views
 - 7. Partial Views
 - 8. View Models
 - 9. HTML and URL Helpers
 - 10. Tag Helpers
 - 11. View Components
 - 12. Client-Side Dependencies
 - 13. Microsoft Library Manager (LibMan)
 - 14. Razor Pages
- 9. HTML Forms
 - 1. Introduction
 - 2. Form Tag Helper
 - 3. Form Submissions
 - 4. Model Binding
- 10. Data Validation
 - 1. Introduction
 - 2. Data Annotations
 - 3. Model Binding
 - 4. Input Tag Helpers
 - 5. Validation Tag Helpers
- 11. Application State

- 1. Introduction
 - 2. HttpContext.Items
 - 3. Session State
 - 4. TempData
- 12. Authentication
 - 1. Introduction
 - 2. ASP.NET Core Identity
 - 3. Cookie Middleware
 - 4. Authorization
 - 5. Claims-Based Authorization
- 13. Error Handling
 - 1. Best Practices
 - 2. HTTP Error Status Codes
 - 3. Status Code Pages
 - 4. Developer Exception Page
 - 5. Exception Filters
- 14. Logging
 - 1. Introduction
 - 2. Configuration
 - 3. ILogger
- 15. Testing
 - 1. Introduction
 - 2. Unit Testing
 - 3. xUnit
 - 4. Testing Controllers
 - 5. Integration Testing
- 16. Web APIs
 - 1. Introduction
 - 2. Retrieval Operations
 - 3. Create Operations
 - 4. Update Operations
 - 5. Delete Operations
 - 6. Bad Requests
 - 7. Cross-Origin Resource Sharing (CORS)
- 17. Using Docker
 - 1. Advantages of Containerized Applications
 - 2. Docker Fundamentals
 - 3. Microsoft ASP.NET Core Docker Images
 - 4. Running a Container
 - 5. Visual Studio Support
 - 6. AWS and Azure
- 18. Deployment
 - 1. Page and View Compilation
 - 2. Publishing
 - 3. Kestrel as an Edge Server
 - 4. IIS as a Reverse Proxy
- 19. Conclusion

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this ASP.NET class:

- Experience with the C# programming language and object-oriented programming concepts.
- Some knowledge of HTML, CSS, and JavaScript concepts.