



ASP.NET Core 3 Development

Course ID #: ASPNC-104

Hours: 28

Course Content

Course Description:

ASP.NET Core 3 is the latest release of Microsoft's cross-platform version of its ASP.NET Web development framework. This ASP.NET Core 3 Development training introduces attendees to ASP.NET Core development and teaches the skills necessary to design and build well-architected web and service-oriented applications that follow the MVC design pattern.

At Course Completion:

After completing this course, student will be able to:

- Understand the goals and benefits of the .NET Core platform
- Learn to make good decisions about application architecture and the choice of data access technology
- Use ASP.NET Core's routing system to achieve a REST-style architecture
- Learn how to build a compelling and maintainable HTML user interface using the Razor view engine and client-side JavaScript
- Gain experience using Blazor, Razor Pages, SignalR, and gRPC
- Learn how to extend and modify ASP.NET Core by creating custom components and templates
- Understand the different cross-platform deployment options available including via Docker containers

Target Student:

Web Developers

Prerequisites:

Previous experience developing web-based applications with C#. Some familiarity with HTML, CSS, and JavaScript.

Topics:

Introduction

- What is .NET Core?
- .NET Core vs. Full Framework
- Overview of ASP.NET Core

.NET Core SDK

- Installation
- Version Management
- Command-Line Interface (CLI)
- Hello World Application
- ASP.NET Core Application Architecture



ASP.NET Core 3 Development

Course ID #: ASPNC-104

Hours: 28

- NuGet Packages and Metapackages
- Application Startup
- Hosting Environments
- Middleware and the Request Processing Pipeline
- Services and Dependency Injection
- MVC vs. Razor Pages

Application Configuration

- Configure and Configure Services
- Configuration Providers and Sources
- Configuration API
- Options Pattern
- HTTPS, GDPR, and HTTP/2

Request Routing

- RESTful Services
- Endpoint Routing
- Attribute-Based Routing
- Route Templates
- Route Constraints

Models

- Persistence Ignorance
- Object-Relational Mapping
- Entity Framework (EF) Core
- Dapper ORM

Controllers

- Responsibilities
- Requirements and Conventions
- Dependencies
- Action Results

Views

- Responsibilities
- Conventions
- Razor Syntax
- Layouts
- View Data and Strongly-Typed Views
- Partial Views
- View Models
- HTML and URL Helpers
- Tag Helpers
- View Components

- Client-Side Dependencies
- Microsoft Library Manager (LibMan)
- Razor Pages

HTML Forms

- Form Tag Helper
- Form Submissions
- Model Binding

Data Validation

- Introduction
- Data Annotations
- Model Binding
- Input Tag Helpers
- Validation Tag Helpers

Blazor

- Server-Side Blazor vs. Client Side Blazor
- Browser Support
- Forms and Validation
- Routing

Application State

- Client-Side vs. Server-Side
- HttpContext.Items
- Session State
- TempData

Authentication

- Introduction
- ASP.NET Core Identity
- Cookie Middleware
- Authorization
- Claims-Based Authorization

Error Handling

- Best Practices
- HTTP Error Status Codes
- Status Code Pages
- Developer Exception Page
- Exception Filters

Logging

- Configuration
- ILogger
- Serilog and Seq
- Testing



ASP.NET Core 3 Development

Course ID #: ASPNC-104

Hours: 28

- Unit Testing
- xUnit
- Testing Controllers
- Integration Testing

Web APIs

- Introduction
- CRUD Operations
- Bad Requests
- Cross-Origin Resource Sharing (CORS)

Remote Procedure Calls (RPC)

- Real-Time Apps with SignalR
- SignalR Clients
- gRPC Services
- gRPC vs. HTTP Services

Using Docker

- Advantages of Containerized Applications
- Docker Fundamentals
- Microsoft ASP.NET Core Docker Images
- Running a Container
- Visual Studio Support
- AWS and Azure

Deployment

- Page and View Compilation
- Publishing
- Load Balancers and Reverse Proxies