

Introduction to ASP.NET Core Development

Duration: 4 Days · **Training format:** Virtual Classroom / Online

Description

ASP.NET is an open-source web framework developed by Microsoft that is used to build web apps and services with .NET, and to build simple, fast and scalable websites based on HTML5, CSS, and JavaScript. Webucator offers a wide variety of ASP.NET courses for both new and experienced developers looking to build web applications using ASP.NET.

ASP.NET Core 2.1 is the latest release of Microsoft's cross-platform version of its ASP.NET Web development framework. Version 2.1 includes several new features and improvements. This 4-day course 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.

Delegates will learn how to

- Design and build applications that follow the MVC design pattern.

Outline

Introduction

- What is .NET Core?
- .NET Core vs .NET 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

- NuGet Packages and Metapackages
- Application Startup
- Hosting Environment

- Middleware and the Request Processing Pipeline
- Services and Dependency Injection

Application Configuration

- Configure Method
- ConfigureServices Method
- MVC Components
- Configuration Providers and Sources
- Configuration API
- Options Pattern

Request Routing

- RESTful API
- Routing Middleware
- Route Templates
- Route Constraints
- MVC Middleware
- Attribute-Based Routing

Models

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

Controllers

- Introduction
- Requirements and Conventions
- Dependencies
- Action Results

Views

- Introduction
- Conventions
- Razor Syntax
- Layouts
- ViewData and ViewBag
- 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

- Introduction
- Form Tag Helper
- Form Submissions
- Model Binding

Data Validation

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

Application State

- Introduction
- 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

- Introduction
- Configuration
- ILogger

Testing

- Introduction
- Unit Testing
- xUnit
- Testing Controllers
- Integration Testing

Web APIs

- Introduction
- Retrieval Operations
- Create Operations
- Update Operations
- Delete Operations
- Bad Requests
- Cross-Origin Resource Sharing (CORS)

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
- Kestrel as an Edge Server
- IIS as a Reverse Proxy

Prerequisites

Knowledge of basic C#