Course Outline

ASP.NET Core 2.0 Development

Course Summary

Description

The training session will cover ASP.NET Core 2.0 and Visual Studio 2017. We take a detailed look at the latest APIs and development techniques for creating dynamic, modular, and testable web sites. We'll explore the new features available in ASP.NET Core and show how to get the best out of the latest toolset and integration possibilities

From the attached lab list, up to five exercises per subject can be chosen. A full lab can take 75 to 100 minutes or longer with a maximum of 200 minutes, or about 2 ½ hours.

Topics

- ASP.NET Core principles
- MVC architecture and benefits
- Creating views, view components, and tag helpers
- Using ASP.NET dependency injection and configuration
- Test-driven development
- Navigation and state management
- Creating RESTful services using Web API
- Creating Single Page Applications (SPAs)
- ASP.NET Identity security

Audience

This course is designed for those wanting to learn ASP.NET Core 2.0 and Visual Studio 2017.

Prerequisites

Before taking this course, you should have the following skills:

- Six months experience of C# programming
- Six months experience with Visual Studio 2015 or 2017
- GitHub development experience is advantageous but not essential
- ASP.NET development experience is advantageous but not essential

Duration

Four days

Course Outline

ASP.NET Core 2.0 Development

Course Outline

I. Introduction to ASP.NET Core
A. Introduction to .NET Core
B. ASP.NET Core essentials
C. Creating a simple ASP.NET Core web app
D. A closer look at ASP.NET Core middleware
E. Creating custom OWIN middleware components

II. Getting Started with MVC
A. Introduction to ASP.NET Core MVC
B. Understanding controllers
C. Understanding views
D. Layout pages
E. A closer look at Razor syntax
F. Razor pages

III. Creating a Complete ASP.NET MVC Application
A. Design considerations
B. Defining models and views
C. Handling form submissions
D. Additional techniques
E. Asynchronous action methods

IV. Tag Helpers
A. Introduction to tag helpers
B. UI tag helpers
C. Link and script tag helpers
D. Environment tag helper
E. Cache tag helper

V. Structuring ASP.NET Core MVC Applications
A. Design considerations
B. Defining the domain model
C. Defining the Web application

VI. Dependency Injection, Configuration, and Entity Framework
A. Dependency injection principles
B. DI in ASP.NET Core
C. Configuration in ASP.NET Core
D. Using Entity Framework Core

VII. Test Driven Development with ASP Core MVC
A. TDD principles
B. Unit testing frameworks for .NET Core
C. XUnit.net walkthrough
D. Unit testing ASP.NET MVC controllers
E. Mocking

VIII. Defining Custom Tag Helpers
A. Custom tag helpers
B. Tag attributes
C. Additional techniques
D. Worked example

IX. Implementing Navigation
A. Defining view-model classes
B. Implementing data filtering in a controller
C. Understanding the routing mechanism
D. Adding custom entries to a route table
E. Defining defaults, parameters, and validation
F. Generating URLs and hyperlinks
G. Custom route constraints

X. State Management
A. Using hidden fields
B. Session and application state
C. Custom model bindings
D. Distributed caching

XI. Creating RESTful Services using Web API
A. Overview of Web API
B. Building servers and clients
C. Content negotiation
D. Attribute routing
E. Custom model binding
F. Invoking RESTful services from Ajax clients

XII. Creating Single Page Applications
A. Overview of SPAs
B. Using GruntJS, NPM, and Bower support
C. Creating SPAs using Angular 4 and Knockout

Course Outline

ASP.NET Core 2.0 Development

Course Outline (cont'd)

XIII. Security using ASP.NET Identity

A. Security concepts
B. Overview of ASP.NET Identity
C. Customization possibilities using ASP.NET Identity

XIV. Proposed Labs

Basic Labs to support some of the Functions Learned: (Note: Pick up to five exercises per subject).

A. Create an ASP.NET Core MVC web app
   1. Get started
   2. Add a controller
   3. Add a view
   4. Add a model
   5. Work with SQL Server LocalDB
   6. Controller methods and views
   7. Add search
   8. Add a new field
   9. Add validation
   10. Examine the Details and Delete methods

B. ASP.NET Core MVC with EF Core:
   1. Create, Read, Update, and Delete operations
   2. Sorting, filtering, paging, and grouping
   3. Migrations
   4. Create a complex data model
   5. Reading related data
   6. Updating related data
   7. Handle concurrency conflicts
   8. Inheritance
   9. Advanced topics

C. Author Tag Helpers in ASP.NET Core
   1. Create Tag Helpers
   2. A minimal Tag Helper
   3. SetAttribute and SetContent
   4. Pass a model to a Tag Helper
   5. Condition Tag Helper
   6. Inspect and retrieve child content

D. Introducing view components
   1. Creating a view component
   2. Invoking a view component
   3. Invoking a view component as a Tag Helper

E. Create Core apps using dotnet watch
   1. Creating/Running .NET Core CLI commands using dotnet watch
   2. Making changes with dotnet watch
   3. Running tests using dotnet watch
   4. Creating dotnet-watch in GitHub

F. Create a Razor Pages web app: (Optional)
   1. Get started with Razor Pages
   2. Add a model to a Razor Pages app
   3. Scaffolded Razor Pages
   4. Work with SQL Server LocalDB
   5. Updating the pages
   6. Add search
   7. Add a new field
   8. Add validation
   9. Upload files