

Progressive Web Application Development using .NET Web API, Entity Framework Core, and Blazor

Download Whitepaper: Accelerate Your Modernization Efforts with a Cloud-Native Strategy
Get Your Free Copy Now

Course Number: WA3064

Duration: 4 days

Overview

Course Description

This 3-day intensive course, which includes both theoretical learning and actionable labs, enables students to apply Blazor to their workplace. Key areas of focus include creating modern Progressive Web Applications with Blazor, understanding the principles of Blazor components, and enabling PWA functionality in existing and new Blazor applications.

Skills Gained

In this training, attendees will learn how to:

- Install and use the Entity Framework package to work with databases on the server-side
- Install packages required to develop the client-side Blazor applications
- Use components, service & dependency injection, layout, and routing

- Enable the Progressive Web Apps features in an application by creating service workers
- Use the Progressive Web Apps features, such as push notifications and storage with IndexedDB

Prerequisites

Attendees should have familiarity with .NET and C#. Basic knowledge of JavaScript is useful but not necessary.

Audience

Course Details

Outline

Chapter 1. Introduction to Blazor, Entity Framework Core, and Progressive Web Apps

- What is Microsoft Blazor?
- What is WebAssembly?
- Blazor WebAssembly
- Server-Side Blazor
- What is Entity Framework Core?
- What are Progressive Web Apps?
- PWA Technologies
- Technologies used in Blazor Apps
- C# in a Blazor App
- EF Core and LINQ
- Blazor vs. Angular or React
- Summary

Chapter 2. Installing Packages and Working with Git

- Installing Blazor
- Installing Entity Framework Core
- Installing ODP for Oracle
- Working with Git on Azure DevOps Services
- Creating a DevOps Project
- Code Repositories
- Creating a Local Repository
- Pushing a Visual Studio Project to DevOps

- What is Git?
- Committing Changes in Git
- Pushing Changes to Azure DevOps
- What is a Branch?
- Creating Branches
- Undoing Changes
- Creating Pull Requests
- Summary

Chapter 3. Getting Started with Entity Framework Core

- Entity Framework Core Overview
- The Entity Framework Model
- The Database Context Object
- Change Tracking
- Installing Entity Framework Core
- Modelling using Database First
- Database First Options
- Navigation Properties
- Modelling using Code First
- Mapping Tables and Fields
- Defining Keys
- Marking Navigation Properties
- Using the Fluent API
- Mapping Tables and Fields
- Defining Keys
- Custom Configuration Classes
- Executing Stored Procedures
- Disposing of the Context
- Summary

Chapter 4. Querying and Saving Data with Entity Framework Core

- What is LINQ?
- Basic Queries with LINQ
- Filtering

- Sorting
- LINQ Function Syntax
- Filtering using Function Syntax
- Sorting using Function Syntax
- Working with Query Results
- Deferred Execution
- Progressively Building a Query
- Forcing Query Resolution
- Retrieving Single Items From a Query
- Calling Stored Procedures
- Using Include and Then Include
- Controlling the Tracking Behavior
- Saving Changes
- Performing Bulk Insert
- Implementing Transactions
- Summary

Chapter 5. Getting Started with Blazor

- Creating a Blazor Hello World Application
- Server-Side Blazor
- Blazor Web Assembly
- Static Hosting
- ASP.NET Core Hosting
- Blazor Fundamentals – Program.cs
- Blazor Fundamentals – wwwroot Folder
- Blazor Fundamentals – Index.html
- Blazor Fundamentals – Blazor.WebAssembly.js
- Blazor Fundamentals – App.razor
- Blazor Fundamentals – MainLayout.razor
- Razor Pages
- Razor Components
- Razor Syntax
- Summary

Chapter 6. Working with Blazor Components

- What are Components?
- What is a routable component?
- Working with Components in Blazor
- Using C# in Components
- Code block within the .razor file
- Using a code-behind file
- Parameterize Components
- Passing Parameters to Child Components
- Getting Parameters from Route Data
- Understanding Component Life Cycle
- Binding – One-way Data Binding
- Binding – Two-way Data Binding
- Two-way Binding with Components
- Binding – Event Binding
- Summary

Chapter 7. Advanced Blazor Component Concepts

- HTML Encoding
- Render Raw HTML
- The Problem with Parameters
- Cascading Parameters
- Cascading by Type
- Flow of Content in Component Hierarchy
- Render Child Content
- Using RenderFragment
- Multiple RenderFragments
- Blazor's RenderTree
- Using RenderTreeBuilder
- Why You Should Hard-code Sequence Numbers
- Rendering Performance and Loops
- Using @key for more Efficient Updates
- Passing Arbitrary Attributes
- Summary

Chapter 8. Layout and Routing

- Layout in Razor Applications
- Creating a Master Layout
- Specifying the Default Layout
- Using a Different Layout for some Pages
- Nested Layouts
- How Routing Works
- Implementing Routing
- Matching a Route
- When no Route is Matched
- Defining Routes
- Route Parameters
- Constraining Route Parameters
- Catch-all Route Parameters
- Navigating via HTML
- Navigating via Code
- Detecting Navigation Events
- Summary

Chapter 9. Forms and Validation

- EditForm Component
- Validation Requirements - Summary
- The Form Model
- Sample Model
- Sample Razor
- Form Components
- Using a Forms Component
- Form Submission
- Forms Validation
- Updating the Model for Validation
- Highlighting Invalid Fields
- Displaying validation Error Message Summary
- Displaying Validation Error Message For Each Field
- Handling Validity in Form Submission
- OnValidSubmit and OnInvalidSubmit Events

- OnSubmit Event
- Summary

Chapter 10. Dependency Injection & JSInterop

- Service and Dependency Injection
- Creating a Service and using Dependency Injection
- Creating a Service
- Register the Injectable Service
- Inject the Service and Use it
- A CRUD-based Service Sample
- Dependency Scopes
- Transient Dependencies
- Singleton Dependencies
- Scoped Dependencies
- Scoped Dependencies in Blazor Server App
- Scoped Dependencies in Blazor WebAssembly
- JavaScript Interop
- Calling a JavaScript Function from C#
- Calling a C# Function from JavaScript
- Summary

Chapter 11. Debugging and Deploying a Blazor Application

- Debugging a Blazor Application
- Debugging Features
- Debugging Limitations
- Ensure Debugging is Enabled
- Debugging using Visual Studio
- Debugging using Chrome/Microsoft Edge
- Debugging using Chrome/Microsoft Edge (contd.)
- Blazor Hosting Models
- Blazor WebAssembly
- Blazor WebAssembly - Pros
- Blazor WebAssembly - Cons
- Blazor Server

- Blazor Server - Pros
- Blazor Server - Cons
- Deploying Blazor Applications
- Publish Locations
- Deployment on IIS
- Deployment on IIS (contd.)
- Summary

Chapter 12. Best Practices for Progressive Web Applications

- View Logic Separation
- Sample View Logic
- Sample View
- Blazor Class Library
- Creating a Blazor Class Library Project
- Sample Model
- Sample Component
- Add Razor Class Library Project Reference
- Using the Custom Component
- Upgrade from .NET Core 3.x to .NET 5
- Optimize Rendering Speed
- Avoid Unnecessary Rendering of the Component Subtree
- ShouldRender
- Component Virtualization
- Component Virtualization (contd.)
- Avoid Creating Granular Components
- Avoid Creating Granular Components (contd.)
- Reusable Components using RenderFragments
- Reusable Components using RenderFragments (contd.)
- Minimize App Download Size
- Summary

Chapter 13. Progressive Web Apps Overview

- Progressive Web Apps (PWA) Overview
- PWA Apps Features

- PWA Features (contd.)
- Comparing PWA to Other Apps
- Business Benefits of PWA Applications
- Characteristics of PWA Applications
- Capable
- Reliable
- Installable
- What is a Service Worker
- Push Notifications
- PWA App Shortcuts
- Basic PWA Features in a Blazor Application
- Enable the PWA Features in an Existing Application
- Enable the PWA Features in an Existing Application (contd.)
- Sample manifest.json
- Implementing Push Notifications
- Developer Tools Tips
- Uninstall PWA Applications
- Summary

Chapter 14. Push Notifications and Storage

- Data Storage
- Web Storage API
- Web Storage API Options
- Web Storage Programming Interface
- Web Storage Examples
- Storing JavaScript Objects
- IndexedDB
- IndexedDB Terminology
- IndexedDB with JavaScript
- IndexedDB with .NET Package
- Important Classes
- Registering IndexedDB Services and Injecting Dependencies
- Important Methods
- Summary

Lab Exercises

- Lab 1. Working with Git on Azure DevOps Services
- Lab 2. Getting Started with Entity Framework Core
- Lab 3. Querying and Saving Data with Entity Framework Core
- Lab 4. Getting Started with Blazor
- Lab 5. Working with Blazor Components
- Lab 6. Advanced Blazor Components
- Lab 7. Layout and Routing
- Lab 8. Creating a Form using EditForm
- Lab 9. Implementing Form Validation
- Lab 10. Implementing Service and Dependency Injection
- Lab 11. Debugging a Blazor Application
- Lab 12. Getting Started with Progressive Web Apps (PWA)
- Lab 13. Adding the IndexedDB Support to an Application
- Lab 14. Push Notifications