

Blazor Server

Course Summary

Description

This course provides an in-depth overview of Microsoft's Blazor Server framework for building interactive client-side web applications with .NET. It begins by explaining the benefits of using Blazor Server. Then, through a combination of both demonstrations and labs, the course proceeds to teach the student how to implement and secure a single-page web application using Blazor Server. This course is intended for web developers who are interested in developing web applications using C# on the client instead of JavaScript.

Objectives

By the end of this course, students will be able to:

- Build a Single Page Application (SPA)
- Explain the Difference Between Blazor WebAssembly and Blazor Server
- Explain the lifecycle of a Blazor Server page
- Configure Identity Server
- Understand debugging on both the server and the client
- Call existing JavaScript libraries
- Store data on the browser
- Configure a Progressive Web App
- Update data on SQL Server
- Share Razor Components between projects

Topics

- Blazor Server Fundamentals
- Hosted Applications
- Authentication and Authorization
- Razor Components
- Data Entry
- Component Class Libraries
- JavaScript Interop
- State Management
- Migrate from .NET Framework to .NET Core
- Advanced Topics

Audience

This course is intended for those responsible for implementing fast, secure web applications that do not rely on JavaScript.

Prerequisite

- Basic understanding of programming principles
- Familiarity with Microsoft Visual Studio
- Experience with C# and HTML

Duration

Five Days

Blazor Server

Course Outline

- I. Blazor Server Fundamentals**
 - A. Overview
 - B. Getting Started
 - C. Project Structure
 - D. Routing
 - E. Default Layout
 - F. Nested Layouts
 - G. Navigation
 - H. Components
 - I. Razor Syntax
 - J. Introduction to Bootstrap
- II. Hosted Applications**
 - A. Benefits
 - B. Solution Structure
 - C. Debug Server and Client
 - D. Error Logging
 - E. Dependency Injection
 - F. HttpClient and JSON Helpers
 - G. Partial Class Support
 - H. SignalR
- III. Authentication and Authorization**
 - A. Overview
 - B. Introduction to Identity Server
 - 1. Registration Page
 - 2. Login Page
 - 3. JSON Web Tokens
 - C. Authorize Attribute
 - D. AuthorizeView Component
 - E. Custom Claims
- IV. Razor Components**
 - A. Overview
 - B. Component Parameters
 - C. Event Handling
 - D. Lifecycle Methods
 - E. Templated Components
 - F. @key Directive
 - G. Attribute Splatting
 - H. Composing Components
 - I. Passing Data Between Components
- V. Data Entry**
 - A. Form Overview
 - B. EditForm Component
 - C. Standard Input Components
 - D. Data Binding
 - E. Data Validation
- VI. Component Class Libraries**
 - A. Getting Started
 - B. Share Components
 - C. Decompile Components
 - D. RenderTreeBuilder Method
 - E. Component State Changes
 - F. Create NuGet Package
- VII. JavaScript Interop**
 - A. Overview
 - B. Invoke JavaScript Functions from .NET
 - 1. Detecting Prerendering
 - 2. Reference Elements
 - 3. Harden JS interop calls
 - 4. InvokeVoidAsync
 - 5. InvokeAsync
 - C. Call Existing JavaScript Libraries
 - D. Invoke .NET Methods from JavaScript
 - 1. Static Method Call
 - 2. Instance Method Call
 - 3. Component Instance Methods
- VIII. State Management**
 - A. Application State
 - B. Server-side
 - C. Circuits & SignalR
 - D. Large Data Transfers
 - E. Using EF Core
 - F. Local Storage
 - G. Session Storage
 - H. Protected Browser Storage

Blazor Server

Course Outline (cont'd)

- IX. *Migrate to .NET Core***
 - A. What is .NET Core?
 - B. What is ASP.NET Core?
 - C. Migrating from .NET Framework
 - D. Migrating from ASP.NET Framework
 - E. Application Startup
 - F. Hosting Environments
 - G. Middleware and the Request Pipeline
 - H. Services and Dependency Injection
 - I. Application Configuration

- X. *Advanced Topics***
 - A. Content Security Policy
 - B. Localization
 - C. Globalization
 - D. Deployment
 - E. Performance Tips
 - F. 3rd Party Components
 - G. Additional Resources

Blazor Server

Labs

Day 1

- Create a simple Blazor Server application
- Create a hosted Server application with a SQL Server backend that will be used for the rest of the labs

Day 2

- Implement authentication and authorization by configuring Identity Server on the application
- Add components to the application such as a modal dialog box and a custom templated component

Day 3

- Add forms to be enable data to be edited on SQL Server using Entity Framework
- Create a Component Class Library to share components

Day 4

- Invoke JavaScript methods from the application
- Manage application state using a service and store data on the browser

Day 5

- Convert the application into a Progressive Web App
- Add localization, globalization and 3rd party controls to the application