



# Building Blazor Applications

**Course #:** BL-210      **Duration:** 3 days

## Prerequisites

Experience with the C# programming language is required, along with familiarity with HTML and CSS.

## Details

This course teaches developers how to create rich-client single-page and multi-page web applications that use the C# language and the full power of the .NET framework executing in the browser.

Students will learn how to integrate client-side and server-side logic and re-use resources and models between both the front-end and back-end of the web application. This course also shows how to integrate client-executing Blazor code with JavaScript to take advantage of the strengths of both languages.

## Software Needed

The required software is Visual Studio 2019 or higher. The free Visual Studio Community 2019 or higher may be used. The operating system should be Windows 7 SP1 or later. Several free tools are also used. These can all be downloaded from the Web.

## Outline

- **Introduction to Blazor**
  - What is Blazor?
  - Blazor Web Assembly vs. Blazor Server
  - How Blazor fits into the Web Ecosystem
  - Setting up Dev Environment
  - Creating a Blazor Application
  - Project Structure and Key Files
  - Running and Debugging the Application
- **Components and Razor Syntax**
  - Creating and Using Components
  - Razor Syntax
  - Passing Parameters
  - Property Binding
  - Render Fragment
  - One-Way Data Binding
  - Two-Way Data Binding
  - Handling Events
- **Deeper Look at Components**
  - Working with Models
  - Lifecycle Methods

- Using Forms and Input Controls
  - Built-in Form Validation
- **Styling Blazor Components**
  - Layouts
  - Global styles
  - CSS Isolation
  - Shadow-Piercing
- **Routing and Navigation**
  - Defining and Using Routes
  - Route Parameters
  - Constraining Route Parameters
  - Navigation via Code
  - Not Found Content
- **Services and Dependency Injection**
  - Dependency Injection in Blazor
  - Creating and Registering Services
  - Creating a RESTful API
  - Using Entity Framework Core
  - Using HttpClient to consume APIs
  - Create/Update/Delete Actions
- **State Management**
  - Server Storage
  - Database Access with Entity Framework
  - Client Storage with LocalStorage
- **JavaScript Interop**
  - Calling JavaScript from Blazor
  - Calling .NET From JavaScript
  - Use Cases and Best Practices
- **Component Communication Patterns**
  - Parent-child Communication
  - Event Callbacks
  - Cascading Parameters
  - Communication Through Services
- **Advanced Forms and Navigation**
  - Using EditContext
  - Custom Validation
  - Using Custom CSS
  - Disabling the Submit Button
- **Unit Testing Blazor Applications**
  - Using bunit with MSTest
  - Creating a Custom TextContext
  - Verifying Rendered Content
  - Mocking JSInterop
  - Triggering Event Handlers
  - Injecting Services
  - Verifying Component State
- **Publishing and Deployment**
  - Publishing Blazor Web Assembly
  - Publishing Blazor Server
  - Hosting Options
  - Performance and Optimization Tips
  - Deploying to Azure App Service
  - Deploying to IIS
  - Continuous Delivery with Azure DevOps