



# 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 may be used. The operating system should be Windows 7SP1 or more recent. Several free tools are also used. These can all be downloaded from the Web.

## Outline

- **Introduction**
  - What is Blazor?
  - Blazor Web Assembly
  - Blazor Server-Side
  - Blazor Development Environment
- **Razor Syntax**
  - Syntax Overview
  - Working with Models
  - Using Classes
  - Control Structures
  - Using Markupstrings
- **Blazor Components**
  - Creating Components
  - Using Components
  - Lifecycle Methods
  - Passing Parameters
- **Advanced Components**
  - Working with Events
  - Data Binding

- Event Callbacks
- RenderFragment
- Generic Fragments
- **Application Infrastructure**
  - Dependency Injection
  - Partial Classes
  - Layout
  - Invoking JavaScript from C#
  - Invoking C# from JavaScript
  - Component References
  - Debugging WebAssembly Apps
- **Building Forms**
  - EditForm and InputText
  - Validation using Data Annotations
  - Building Filter Components
  - Building Selector Components
  - Reordering List Elements
  - Building AutoComplete Components
- **Server Communications**
  - Creating a RESTful API
  - Using Entity Framework Core
  - Using the HttpClient Service
  - Retrieving Data
  - Pagination
  - Create/Update/Delete Actions
- **Security Basics**
  - Exploring the Authentication State Provider
  - Using AuthorizeView
  - Protecting Components with Authorize
  - Reading Authentication State
  - Configuring Identity
- **Applying Security**
  - Building an Authentication State Provider
  - Building Registration, Login and Logout Components
  - Implementing Roles
  - Renewing Authentication Tokens
  - Automatic Logout when Inactive
- **Razor Class Libraries**
  - Sharing Components
  - Accessing CSS and JavaScript Resources
  - Dependency Injection
  - Routable Components
  - Cascading Parameters
- **Blazor Server Apps**
  - Creating Blazor Server Apps
  - JavaScript in Blazor Server
  - Dual-Architecture
  - Enabling Authentication
  - Avoiding Firing Events Twice
- **Internationalization**
  - What is Internationalization?
  - Multi-Language Blazor WebAssembly
  - Manually Changing Language
  - Culture-Dependent Formats
  - Multi-Language Error Messages
  - Internationalization in Blazor Server Apps
- **Appendix A: Deploying Blazor Applications**
  - Deploying to Azure App Service
  - Deploying to Azure Storage
  - Deploying to IIS

