Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Build Web Apps with Blazor

Class Duration

21 hours of live training delivered over 3 days.

Student Prerequisites

- HTML, CSS, and JavaScript programming experience
- C# programming experience

Target Audience

Web developers with C# experience who want to build interactive web UIs with Blazor instead of a JavaScript framework. Teams that also need APIs, data access, and real-time updates can continue with *Full-Stack Web Apps with Blazor and ASP.NET Core APIs*.

Description

This course teaches ASP.NET Core Blazor for building interactive web UIs in C#. Students build Razor components, bind data, handle events, and pick a render mode for each page. The course covers forms and validation, QuickGrid, routing, and JavaScript interop. It also covers state that survives prerendering and circuit pauses, and passkey sign-in with ASP.NET Core Identity. The last module publishes and hosts a Blazor app.

Learning Outcomes

- Choose the right render mode for each page
- Build composable UIs with Razor components
- Use Razor syntax for data binding and event handling
- Build validated forms with EditForm and asynchronous validation rules
- Display and page through data with QuickGrid and Virtualize
- Configure page routing and navigation
- Integrate client-side JavaScript with Blazor
- Implement passkey authentication with ASP.NET Core Identity
- Deploy a Blazor application to production

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved



To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Students will need permission to install the .NET 10 SDK and the .NET 11 SDK, plus Visual Studio 2026 or Visual Studio Code with the C# Dev Kit, on their computers. Also, students will need permission to install NuGet Packages. If students are unable to configure a local environment, a cloud-based environment can be provided.

Training Topics

Getting Started with Blazor

- What is Blazor?
- Comparison to Other Web Frameworks
- The Blazor Web App Model
- Server-Side & WebAssembly
- LTS and STS .NET Releases

Running Blazor Code

- Single Page Applications
- What is WebAssembly?
- Browser Compatibility
- WebAssembly vs. JavaScript
- How does .NET run in a web browser?

Blazor Application

- Project Templates
- Create a New Application
- Hosting Blazor in an ASP.NET Core Server
- Configuration
- Environment Variables in WebAssembly
- Dependency Injection
- Environments and the EnvironmentView Component
- Logging
- Handling Errors

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

- Debugging WebAssembly in the Browser

Razor Components and Data Binding

- What is a Component?
- Creating a Data Model
- Binding the Data Model to the HTML
- Passing Arbitrary Attributes
- Handling Events
- Manually Triggering Re-rendering

Rendering

- Render Modes
- Static Server Rendering
- Interactive Server Rendering
- Interactive WebAssembly Rendering
- Interactive Auto Rendering
- AOT vs. JIT Compilation
- Streaming Rendering
- Output Caching and the CacheView Component
- TempData and Session State in Static Rendering
- Render Modes and Consuming Data
- Render Modes and JavaScript

Composing Razor Components

- Splitting a Component into Smaller Components
- One-Way Data Binding
- Two-Way Data Binding
- Parent-to-Child Data with Parameters
- Child-to-Parent Data with Event Callbacks
- Refs to DOM Elements
- Razor Component Libraries
- Razor Component Design Patterns
- Parameters are Immutable
- Required Component Parameters
- Lift State Up
- Managing State in General
- Section Components

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

State Persistence and Reconnection

- Persistent Component State
- Prerendering without Loading Data Twice
- Circuit Reconnection
- The ReconnectModal Component
- Pausing and Resuming Circuits
- Server-Triggered Circuit Pauses
- Automatic Pause for Inactive Browser Tabs

Displaying Data with QuickGrid and Virtualize

- QuickGrid Columns, Sorting, and Paging
- URL-Based Paging and Sorting
- Conditional Row Styling with RowClass
- Row Click Handling
- Column Options Menus
- Virtualizing Variable-Height Items
- Scroll Anchoring
- Scrolling to an Item

Razor Component Forms

- What is the Purpose of a Form?
- Form, Input, Select, and TextArea Elements
- Two-Way Data Binding with Form Elements
- The EditForm Component
- The Edit Context
- Specialized Input Components
- Field Labels with the DisplayName Component
- Applying Validation to the Form
- Validation Attributes on the View Model
- Custom Validation Attributes
- Validating Nested Objects and Collections
- Client-Side Validation for Static Forms
- Asynchronous Validation Rules
- Antiforgery Protection

Razor Component Pages and Routing

- What is the Page model?

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

- Razor Pages vs. Razor Components
- Using a Razor Component as a Page
- The Router Component
- Configuring Page Routing
- Routing to Components in Other Assemblies
- Route Parameters and Constraints
- Using the Query String
- Enhanced Navigation for Static Pages
- Not Found Pages
- Relative Navigation
- NavLink Matching

Interacting with JavaScript

- What is JavaScript Interop?
- When is JavaScript Needed?
- Synchronous vs. Asynchronous Calls
- JavaScript Initializers
- Calling JavaScript from a Component
- Calling .NET from JavaScript
- Collocated JavaScript Modules

Authentication, Authorization, and Auditing

- What is Authentication?
- What is Authorization?
- What is Auditing?
- Authentication Models for Blazor
- Passkey Sign-In with ASP.NET Core Identity
- Refreshing Authentication State in Circuits
- Component Attribute-based Authorization
- Authorization Components
- Authorization with C# Code
- Roles and Claims
- Blazor Identity UI

Deploying Blazor Apps

- Publishing Blazor Web Apps
- Publishing Standalone WebAssembly Apps

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved



To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

- WebAssembly Trimming and AOT Compilation
- Hosting on IIS and Linux
- Hosting in Containers and Cloud Services
- Content Security Policy
- OpenTelemetry and Aspire Service Defaults

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved