



Blazor WebAssembly by example : use practical projects to start building web apps with .NET 7, Blazor WebAssembly, and C# /

Wright, Toi B.,
author

Monografía

Blazor WebAssembly helps developers build web applications without the need for JavaScript, plugins, or add-ons. With its continued growth in popularity, getting started with Blazor now can open doors to new career paths and exciting projects - and Blazor WebAssembly by Example will make your first steps easier. This is a project-based guide that will teach you how to build single-page web applications with Blazor, focusing heavily on the practical over the theoretical by providing detailed step-by-step instructions for each project. The author also includes a video for each project showing her following the step-by-step instructions, so readers can use them if they're unsure about any particular step. In this updated edition, you'll start by building simple standalone web applications and gradually progress to developing more advanced hosted web applications with SQL Server backends. Each project will cover a different aspect of the Blazor WebAssembly ecosystem, such as Razor components, JavaScript interop, security, event handling, debugging on the client, application state, and dependency injection. The book's projects get more challenging as you progress, but you don't have to complete them in order, which makes this book a valuable resource for beginners as well as those who just want to dip into specific topics. By the end of this book, you will have experience and lots of know-how on how to build a wide variety of single-page web applications with .NET, Blazor WebAssembly, and C#

<https://rebiunoda.pro.baratznet.cloud:28443/OpacDiscovery/public/catalog/detail/b2FpOmNlbGVicmF0aW9uOmVzLmJhcmF0ei5yZW4vMzMwMDI5MTY>

Título: Blazor WebAssembly by example use practical projects to start building web apps with .NET 7, Blazor WebAssembly, and C# Toi B. Wright and Scott Hanselman

Edición: 2nd ed

Editorial: Birmingham, England Packt Publishing Ltd. [2023] 2023

Descripción física: 1 online resource (439 pages)

Mención de serie: Expert insight

Nota general: Includes index

Contenido: Cover -- Copyright -- Foreword -- Contributors -- Table of Contents -- Preface -- Chapter 1: Introduction to Blazor WebAssembly -- Benefits of using the Blazor framework -- .NET Framework -- Open source -- SPA framework -- Razor syntax -- Awesome tooling -- Supported by Microsoft -- Hosting models -- Blazor Server -- Advantages of Blazor Server -- Disadvantages of Blazor Server -- Blazor Hybrid -- Advantages of Blazor Hybrid -- Disadvantages of Blazor Hybrid -- Blazor WebAssembly -- Advantages of Blazor WebAssembly -- Disadvantages of Blazor WebAssembly -- Hosting model differences -- What is WebAssembly? -- WebAssembly goals -- WebAssembly support -- Setting up your PC -- Installing Microsoft Visual Studio Community Edition -- Installing .NET 7.0 -- Installing Microsoft SQL Server Express -- Create a Microsoft Azure account -- Summary -- Questions -- Further reading -- Chapter 2: Building Your First Blazor WebAssembly Application -- Creating the Demo Blazor WebAssembly Project Technical Requirements -- Razor components -- Using components -- Parameters -- Required parameters -- Query strings -- Naming components -- Component life cycle -- Component structure -- Directives -- Markup -- Code block -- Routing -- Route parameters -- Optional route parameters -- Catch-all route parameters -- Route constraints -- Razor syntax -- Inline expressions -- Control structures -- Conditionals -- Loops -- Hot Reload -- Creating the Demo Blazor WebAssembly project -- Project overview -- Getting started with the project -- Running the Demo project -- Examining the Demo project's structure -- The Properties folder -- The wwwroot folder -- The App component -- The Shared folder -- The Pages folder -- The Client folder -- The _Imports.razor file -- Examining the shared Razor components -- The MainLayout component -- The NavMenu component The SurveyPrompt component -- Examining the routable Razor components -- The Index component -- The Counter component -- The FetchData component -- Using a component -- Modifying a component -- Adding a parameter to a component -- Using a parameter with an attribute -- Adding a route parameter -- Using partial classes to separate markup from code -- Summary -- Questions -- Further reading -- Chapter 3: Debugging and Deploying a Blazor WebAssembly App -- Technical requirements -- Debugging a Blazor WebAssembly.app -- Debugging in Visual Studio -- Debugging in the browser -- Understanding logging -- Understanding log levels -- Setting the minimum log level -- Handling exceptions -- Setting error boundaries -- Creating a custom error component -- Using ahead-of-time (AOT) compilation -- Deploying a Blazor WebAssembly app to Microsoft Azure -- Creating the "guess the number" project -- Project overview -- Getting started with the project -- Adding a Game component -- Adding the code -- Adding a style sheet -- Setting up and playing the game -- Adding logging -- Debugging in Visual Studio -- Updating the code -- Debugging in the browser -- Adding an ErrorBoundary component -- Deploying the application to Microsoft Azure -- Summary -- Questions -- Further reading -- Chapter 4: Building a Modal Dialog Using Templated Components -- Technical requirements -- Using RenderFragment parameters -- Using EventCallback parameters -- Understanding CSS isolation -- Enabling CSS isolation -- Supporting child components -- Creating the modal dialog project -- Project overview -- Getting started with the project -- Adding the Dialog component -- Add a CSS file -- Test the Dialog component -- Add EventCallback parameters -- Add RenderFragment parameters -- Create a Razor class library -- Test the Razor class library -- Add a component to the Razor class library -- Summary Questions -- Further reading -- Chapter 5: Building a Local Storage Service Using JavaScript Interoperability (JS Interop) -- Technical requirements -- Why use JavaScript? -- Exploring JS interop -- Loading JavaScript code -- Invoking a JavaScript function from a .NET method -- InvokeAsync -- InvokeVoidAsync -- IJSInProcessRuntime -- Invoking a .NET method from a JavaScript function -- Using local storage -- Creating the local storage service -- Project overview -- Getting started with the project -- Writing JavaScript to access local storage -- Adding the ILocalStorageService interface -- Creating the LocalStorageService class -- Creating the DataInfo class -- Writing to local storage -- Adding a colocated JavaScript file -- Reading from local storage -- Summary -- Questions -- Further reading -- Chapter 6: Building a Weather App as a Progressive Web App (PWA) -- Technical requirements -- Understanding PWAs -- HTTPS -- Manifest files -- Service workers -- Working with manifest files -- Working with service workers -- Service worker life cycle -- Install -- Activate -- Fetch -- Updating a service worker -- Types of service workers -- Offline page -- Offline copy of pages -- Offline copy with offline page -- Cache-first network -- Advanced caching -- Background sync -- Using the CacheStorage API -- Using the Geolocation API -- Using the OpenWeather One Call API -- Creating a PWA -- Project overview -- Getting started with the project -- Add JavaScript to determine our location -- Invoke the JavaScript function -- Add an OpenWeather class -- Install Bootstrap -- Add a DailyForecast component -- Fetch the forecast -- Display the forecast -- Add the logo -- Add a manifest file -- Add a simple service worker -- Test the service worker -- Install the PWA -- Uninstall the PWA -- Summary -- Questions -- Further reading Chapter 7: Building a Shopping Cart Using Application State -- Technical requirements -- Introducing the application state -- Understanding DI -- DI containers -- Service lifetimes --

Singleton -- Scoped -- Transient -- Creating the shopping cart project -- Project overview -- Creating the shopping cart project -- Installing Bootstrap -- Adding a Store component -- Adding a Product class -- Finishing the Store component -- Demonstrating that the application state has been lost -- Creating the the ICartService interface -- Creating the CartService class -- Registering CartService in the DI container -- Injecting CartService into the Store component -- Adding the cart total to all the pages -- Using the OnChange method -- Summary -- Questions -- Further reading -- Chapter 8: Building a Kanban Board Using Events -- Technical requirements -- Event handling -- Lambda expressions -- Preventing default actions -- Focusing an element -- Attribute splatting -- Arbitrary parameters -- Creating the Kanban board project -- Project overview -- Create the Kanban board project -- Install Bootstrap -- Add the classes -- Create the Dropzone component -- Add a stylesheet -- Create the Kanban board -- Create the NewTask component -- Use the NewTask component -- Summary -- Questions -- Further reading -- Chapter 9: Uploading and Reading an Excel File -- Technical requirements -- Uploading files -- Resizing images -- Handling multiple files -- Using virtualization -- Rendering Local Data -- Rendering Remote Data -- Reading Excel files -- Creating the Excel reader project -- Project overview -- Create the Excel reader project -- Install the Open XML SDK -- Add the Medals class -- Upload the Excel file -- Read the Excel file -- Populate the medals collection -- Render the Medals collection -- Summary -- Questions -- Further reading Chapter 10: Using Azure Active Directory to Secure a Blazor WebAssembly Application -- Technical requirements -- Understanding the difference between authentication and authorization -- Authentication -- Authorization -- Working with authentication -- Controlling the user interface using authorization -- The Authorize attribute -- The AuthorizeView component -- Creating the claims viewer project -- Project overview -- Create the claims viewer project -- Add the application to Azure AD -- Add a user and a group to Azure AD -- Add the required NuGet packages -- Enable authentication -- Add an Authentication component -- Add a LoginDisplay component -- Add a Secure component -- Add a WhoAmI component -- View the JSON Web Token (JWT) -- Add an authentication policy -- Summary -- Questions -- Further reading -- Chapter 11: Building a Task Manager Using ASP.NET Web API -- Technical requirements -- Understanding hosted applications -- Client project -- Server project -- Shared project -- Using the HttpClient service -- Using JSON helper methods -- GetFromJsonAsync -- PostAsJsonAsync -- PutAsJsonAsync -- HttpClient.DeleteAsync -- Creating the TaskManager project -- Project overview -- Create the TaskManager project -- Adding the TaskItem class -- Adding the TaskItem API controller -- Setting up SQL Server -- Install Bootstrap -- Displaying the tasks -- Completing the tasks -- Deleting tasks -- Adding new tasks -- Summary -- Questions -- Further reading -- Chapter 12: Building an Expense Tracker Using the EditForm Component -- Technical requirements -- Creating a data entry form -- Binding a form -- Submitting a form -- Using the built-in input components -- InputCheckbox -- InputDate -- InputFile -- InputNumber -- InputRadio -- InputRadioGroup -- InputSelect -- InputText -- InputTextArea -- Using the validation components -- Locking navigation Creating the expense tracker project

ISBN: 1-80324-753-3

Materia: Application software Web applications- Development

Autores: Hanselman, Scott, author

Enlace a formato fisico adicional: Print version Wright, Toi B. Blazor WebAssembly by Example Birmingham : Packt Publishing, Limited,c2023

Punto acceso adicional serie-Título: Expert insight

Baratz Innovación Documental

- Gran Vía, 59 28013 Madrid
- (+34) 91 456 03 60
- informa@baratz.es