

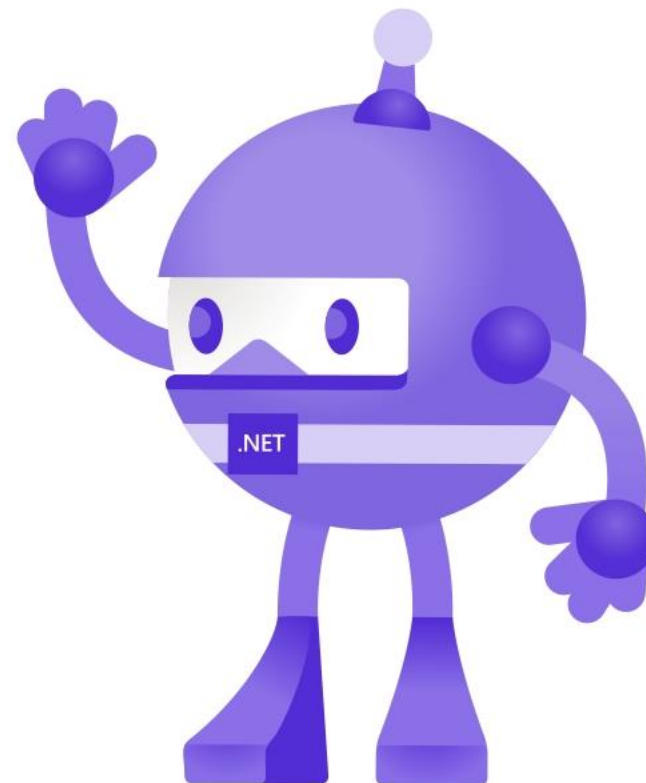
.NET Conference 2024



Full stack web UI con Blazor e .NET 8



Andrea Dottor



SPONSOR



Blazor: Full Stack Web UI framework!

"Blazor is a full-stack web UI framework and is recommended for most web UI scenarios."

Source: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/choose-web-ui>

New project template: **Blazor Web App**



Blazor Web App

A project template for creating a Blazor web app that supports both server-side rendering and client interactivity. This template can be used for web apps with rich dynamic user interfaces (UIs).

C#

Linux

macOS

Windows

Blazor

Cloud

Web

Additional information

Blazor Web App

C#

Linux

mac

Framework ⓘ

.NET 8.0 (Long Term Support)

Authentication type ⓘ

None

☒ Configure for HTTPS ⓘ

Interactive render mode ⓘ

None

Interactivity location ⓘ

Global

☒ Include sample pages ⓘ

☐ Do not use top-level statements ⓘ

Server-side Rendering (SSR)

Blazor is operating like MVC or Razor Pages.



SEO friendly

Additional information

Blazor Web App

C#

Linux

mac

Framework ⓘ

.NET 8.0 (Long Term Support)

Authentication type ⓘ

None

☒ Configure for HTTPS ⓘ

Interactive render mode ⓘ

None

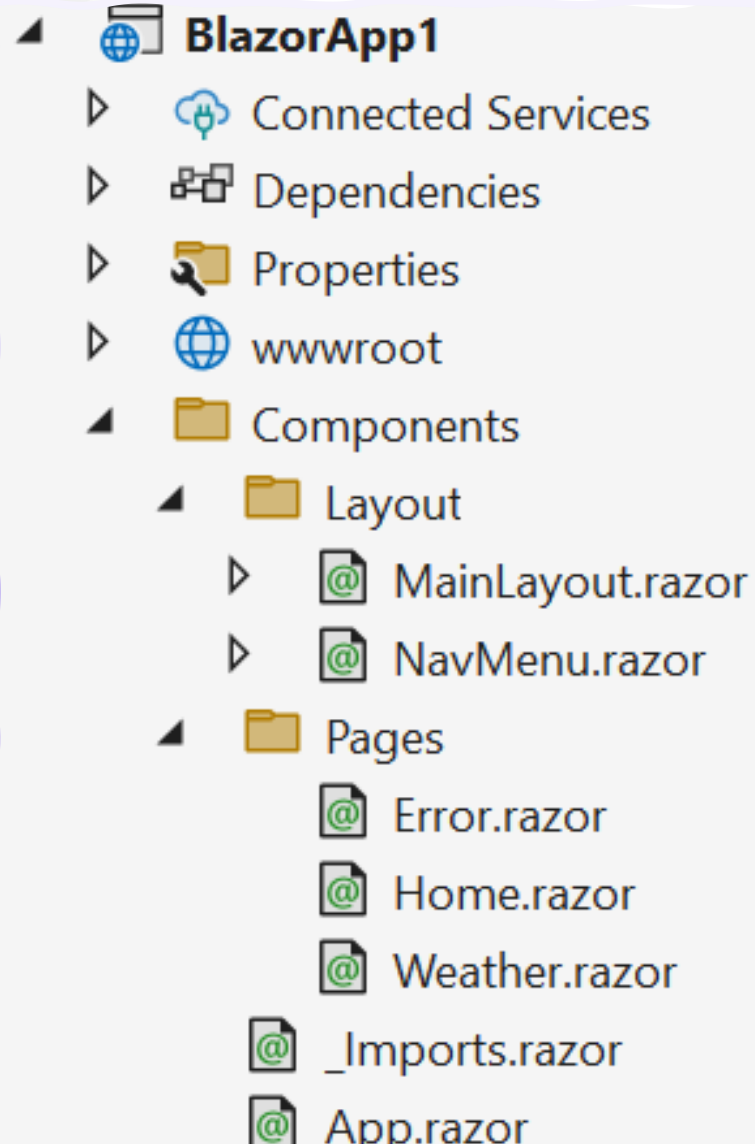
Interactivity location ⓘ

Global

☒ Include sample pages ⓘ

☐ Do not use top-level statements

Server-side Rendering (SSR)



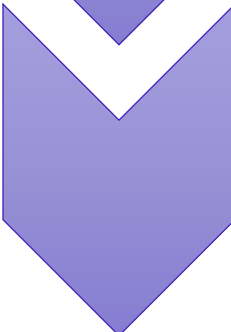
Streaming Rendering



The component initially renders placeholder content ("Loading...") without waiting for the asynchronous delay to complete.



Asynchronous operation are executed



The updated content is streamed to the response and patched into the DOM

Enhanced navigation

Blazor intercepts the request navigation and performs a fetch request instead, rather than a full-page load.
Blazor patches the response content into the page's DOM.



SPA like experience

Enhanced form handling

- Blazor intercepts the form submit and performs a fetch request instead, without fully reloading the page.
- To enable enhanced form handling, add the Enhance parameter to EditForm forms or the data-enhance attribute to HTML forms (<form>)



SPA like experience

DEMO

Interactive render modes



Additional information

Blazor Web App C# Linux macOS

Framework ⓘ

.NET 8.0 (Long Term Support)

Authentication type ⓘ

None

☒ Configure for HTTPS ⓘ

Interactive render mode ⓘ

Server

None

Server

WebAssembly

Auto (Server and WebAssembly)

☐ Do not use top-level statements ⓘ

☐ Enlist in .NET Aspire orchestration ⓘ

InteractiveServer

Use **Blazor Server** for interactive components.



Additional information

Blazor Web App

C#

Linux

macOS

Framework ⓘ

.NET 8.0 (Long Term Support)

Authentication type ⓘ

None

☒ Configure for HTTPS ⓘ

Interactive render mode ⓘ

Server

None

Server

WebAssembly

Auto (Server and WebAssembly)

☐ Do not use top-level statements ⓘ

☐ Enlist in .NET Aspire orchestration ⓘ

InteractiveWebAssembly

Use **Blazor WebAssembly** for interactive components.

Additional information

Blazor Web App C# Linux macOS

Framework ⓘ

.NET 8.0 (Long Term Support)

Authentication type ⓘ

None

☒ Configure for HTTPS ⓘ

Interactive render mode ⓘ

Server

None

Server

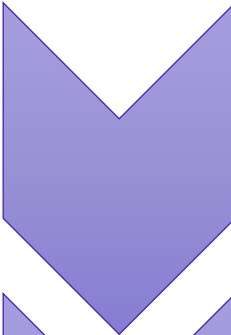
WebAssembly

Auto (Server and WebAssembly)

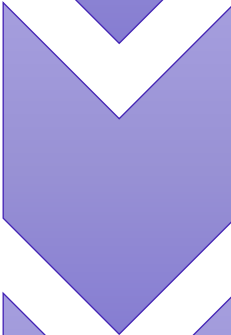
☐ Do not use top-level statements ⓘ

☐ Enlist in .NET Aspire orchestration ⓘ

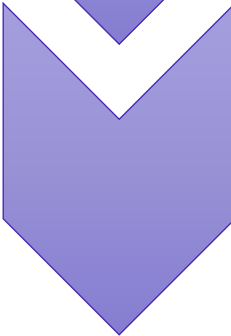
InteractiveAuto



Components initially rendered and use **Blazor Server**.



In background the browser download the Blazor WebAssembly app.



At next request, the components use **Blazor WebAssembly** app.



Additional information

Blazor Web App C# Linux macOS

Framework ⓘ

.NET 8.0 (Long Term Support)

Authentication type ⓘ

None

☒ Configure for HTTPS ⓘ

Interactive render mode ⓘ

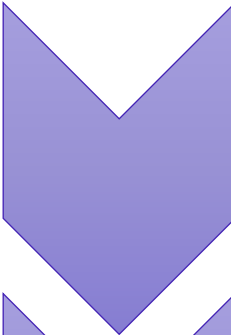
Server

- None
- Server
- WebAssembly
- Auto (Server and WebAssembly)

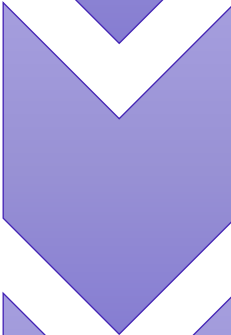
☐ Do not use top-level statements ⓘ

☐ Enlist in .NET Aspire orchestration ⓘ

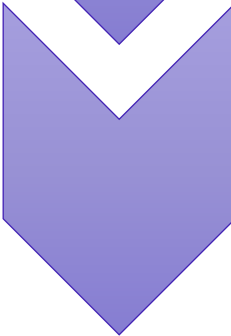
InteractiveAuto



Components initially rendered and use **Blazor Server**.



In background the browser download the Blazor WebAssembly app.



At next request, the components use **Blazor WebAssembly** app.

Interactivity location

Globally (App.razor)

```
<Routes @rendermode="InteractiveServer" />
```

Component instance

```
<Counter @rendermode="InteractiveServer"/>
```

Component definition

```
@page "/counter"  
@rendermode InteractiveServer
```

Additional information

Blazor Web App

C#

Linux

macOS

Windows

Framework [i](#)

.NET 8.0 (Long Term Support)

Authentication type [i](#)

None

☒ Configure for HTTPS [i](#)

Interactive render mode [i](#)

None

Interactivity location [i](#)

Global

Per page/component

Global

☐ Do not use top-level statements [i](#)

Dependency Injection (client/server)

In the server project, in the Program.cs for SSR and InteractiveServer.

If application use InteractiveWebAssembly, in the project *.Client we have a Program.cs for define client dependencies

DEMO

Blazor Identity UI



Additional information

Blazor Web App

C#

Linux

macOS

Windows


Blazor

Cloud

Web

Framework 


.NET 8.0 (Long Term Support)

Authentication type 

None

None


Individual Accounts

Interactive render mode 

None

Interactivity location 

Global

☒ Include sample pages 

☐ Do not use top-level statements 

Additional information

Blazor Web App

C#


Linux

macOS

Windows

Framework 


.NET 8.0 (Long Term Support)

Authentication type 

None

None


Individual Accounts

Interactive render mode 

None

Interactivity location 

Global

☒ Include sample pages 

☐ Do not use top-level statements 

Components

Account

Pages

Manage

@_Imports.razor

@ChangePassword.razor

@DeletePersonalData.razor

@Disable2fa.razor

@Email.razor

@EnableAuthenticator.razor

@ExternalLogins.razor

@GenerateRecoveryCodes.razor

@Index.razor

@PersonalData.razor

@ResetAuthenticator.razor

@SetPassword.razor

@TwoFactorAuthentication.razor

@_Imports.razor

@AccessDenied.razor

@ConfirmEmail.razor

@ConfirmEmailChange.razor

@ExternalLogin.razor

@ForgotPassword.razor

@ForgotPasswordConfirmation.razor

@InvalidPasswordReset.razor

Section

Control the content in a Razor component from a child Razor component.

Sections can be used in both layouts and across nested parent-child components.

```
<SectionOutlet SectionName="top-bar" />
```

```
<SectionContent SectionName="top-bar">
```

```
    <button class="btn btn-primary" @onclick="IncrementCount">Click me</button>
```

```
</SectionContent>
```


Migrate to .NET 8

"Blazor Server apps are supported in .NET 8 without any code changes."

"Blazor WebAssembly apps are supported in .NET 8 without any code changes."

- Update the .NET SDK version in global.json
- Update the target framework to net8.0
- Update package references to .NET 8

Migrate to Blazor Web App (.NET 8)

To optionally adopt all of the new Blazor Web App conventions, we recommend the following process:

- Create a new app from the Blazor Web App project template.
- Move the your app's components and code to the new Blazor Web App app, making modifications to adopt new features.
- Update the layout and styles of the Blazor Web App.

Code and slide

Repository **Umarell-BlazorWebApp**

<https://github.com/andreadottor/Umarell-BlazorWebApp>



Grazie!



@andreadottor

{

name:

email:

x:

linkedin:

}

Andrea Dottor

andrea@dottor.net

@dottor

@andreadottor



.NET Conference 2024



VOTAZIONE

La tua opinione è importante
Aiutaci a migliorare dandoci
il tuo feedback sull'evento

