Rome .NET Conference 2025

Tutte le novità di ASP.NET Core e Blazor in .NET 9

Andrea Dottor
Microsoft MVP Developer Technologies

.NET IN PILLOLE il podcast

Microsoft® Most Valuable Professional

Microsoft BCSoft UNIKEY KEY2 Packt JETBRAINS

Sponsor

Platinum Sponsor

Gold Sponsor

Technical Sponsor

ASP.NET Core in .NET 9

Quality & fundamentals

Developer experience

Cloud native

The fastest ASP.NET Core yet!

- ⬆ **15%** throughput*
- ⬇ **93%** memory usage on high core-count machines*
- ⬇ **25%** startup time for Blazor WebAssembly**

WebSocket message compression for Blazor Server

Precompression and improved caching for static web assets

* TechEmpower JSON Min APIs, Intel Gold 56 cores (logical) Linux
** Lighthouse

Blazor WebAssembly improvements

Investigate Blazor WebAssembly startup performance characteristics

https://github.com/dotnet/aspnetcore/issues/54353

"After introducing a change that utilized the STJ source generator during WebAssembly startup, we noticed a WebAssembly startup time improvement of 15-30%. These numbers may vary for larger apps, or apps that do significant work in user code during startup.

Finding a way to eliminate JSON serialization altogether during startup may result in further improvement of startup time (I would conservatively estimate an additional 15% improvement, possibly higher for Blazor Web scenarios because it performs more serialization on startup)."

Blazor WebAssembly improvements

Investigate Blazor WebAssembly startup performance characteristics

https://github.com/dotnet/aspnetcore/issues/54353

"*After introducing a change that utilized the STJ source generator during WebAssembly startup, we noticed a WebAssembly startup time improvement of 15-30%. These numbers may vary for larger apps, or apps that do significant work in user code during startup.*

Finding a way to eliminate JSON serialization altogether during startup may result in further improvement of startup time (I would conservatively estimate an additional 15% improvement, possibly higher for Blazor Web scenarios because it performs more serialization on startup)."

Blazor Server performance

WebSocket compression enabled by default.

OpenAPI

OpenAPI Document Generation

Built-in support for generating OpenAPI documents representing controller-based or minimal APIs via the **Microsoft.AspNetCore.OpenApi** package.

Adding **builder.Services.AddOpenApi()** and **app.MapOpenApi()** in the app's configuration to enable OpenAPI document generation.

OpenAPI Document Generation

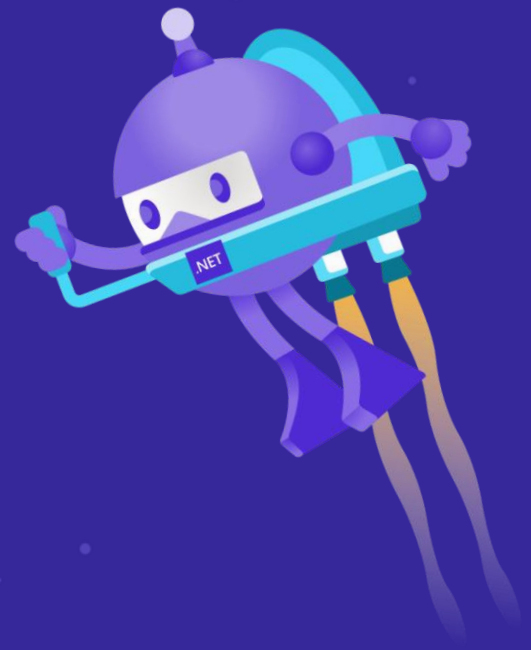
```
var builder = WebApplication.CreateBuilder();

builder.Services.AddOpenApi();

var app = builder.Build();

app.MapOpenApi();

app.MapGet("/hello/{name}", (string name) => $"Hello {name}"!);
app.Run();
```

Generatation at build-time

OpenAPI documents can also be generated at build-time by adding the **Microsoft.Extensions.ApiDescription.Server** package

```
<PropertyGroup>
  <OpenApiDocumentsDirectory>$(MSBuildProjectDirectory)</OpenApiDocumentsDirectory>
  <OpenApiGenerateDocuments>true</OpenApiGenerateDocuments>
</PropertyGroup>
```

Demo



Web UI

Blazor, Razor Pages, MVC

Static asset delivery optimization

MapStaticAssets is a new middleware that helps optimize the delivery of static assets in any ASP.NET Core app, including Blazor apps.

Benefits:

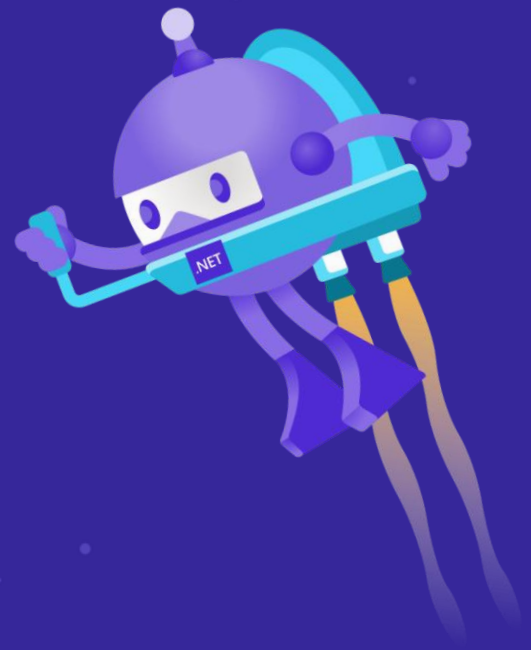
- Build time compression for all the assets in the app:
  - gzip during development and gzip + brotli during publish.
  - All assets are compressed with the goal of reducing the size of the assets to the minimum.
- Content based Etags
  - The ETags for each resource are the Base64 encoded string of the SHA-256 hash of the content. This ensures that the browser only redownloads a file if its contents have changed.

Static asset delivery optimization

File	Original	Compressed	% Reduction
bootstrap.min.css	163	17.5	89.26%
jquery.js	89.6	28	68.75%
bootstrap.min.js	78.5	20	74.52%
Total	**331.1**	**65.5**	**80.20%**

File	Original	Compressed	% Reduction
fluent.js	384	73	80.99%
fluent.css	94	11	88.30%
Total	**478**	**84**	**82.43%**

Demo



Blazor

Detect rendering location, interactivity, and assigned render mode at runtime

New API designed to simplify the process of querying component states at runtime

- Determine the current execution location of the component
- Check if the component is running in an interactive environment
- Retrieve the assigned render mode for the component

Determine the current execution location

RendererInfo.Name returns the location where the component is executing:

- **Static**: On the server (SSR) and incapable of interactivity.
- **Server**: On the server (SSR) and capable of interactivity after prerendering.
- **WebAssembly**: On the client (CSR) and capable of interactivity after prerendering.
- **WebView**: On the native device and capable of interactivity after prerendering.

Check if the component is running in an interactive environment

RendererInfo.IsInteractive indicates if the component supports interactivity at the time of rendering. The value is true when rendering interactively or false when prerendering or for static SSR.

Retrieve the assigned render mode

ComponentBase.AssignedRenderMode exposes the component's assigned render mode:

- **InteractiveServer** for Interactive Server.
- **InteractiveAuto** for Interactive Auto.
- **InteractiveWebAssembly** for Interactive WebAssembly.

Improved Server-Side Reconnection Experience

Immediate Reconnection Attempt

- When a user navigates back to an app with a disconnected circuit, reconnection is attempted immediately rather than waiting for the next reconnect interval.

Automatic Page Refresh

- If a reconnection attempt reaches the server but the server has already released the circuit, a page refresh occurs automatically.

Computed Backoff Strategy

- The first several reconnection attempts occur in rapid succession without a retry interval before computed delays are introduced between attempts.

Modernized Reconnect UI

Add static server-side rendering (SSR) pages to a globally-interactive Blazor Web App

Simplify the addition of static SSR pages to Blazor Web Apps that use global interactivity.

ExcludeFromInteractiveRouting Attribute

- This attribute causes navigation to the page to exit from interactive routing, forcing a full-page reload.

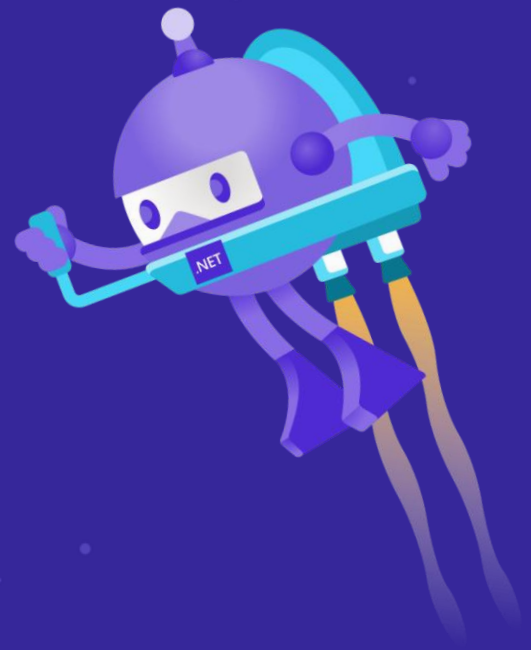
Full-Page Reload

- Inbound navigation performs a full-page reload, allowing the app to switch to a different top-level render mode.

HttpContext.AcceptsInteractiveRouting Extension Method

- Detects whether [ExcludeFromInteractiveRouting] is applied to the current page.

Demo



Blazor

Simplified Authentication State Serialization for Blazor Web Apps

Simplified Authentication State Serialization for Blazor Web Apps

New APIs for Authentication

- Easier to add authentication to existing Blazor Web Apps
- Simplifies the process of serializing and deserializing authentication state

Simplified Code Implementation

- Reduces the amount of code needed to implement authentication state serialization
- Easier to add authentication to existing projects

Improved User Experience

- Ensures authentication state is available during prerendering
- Enhances security by handling authentication on the server

New APIs

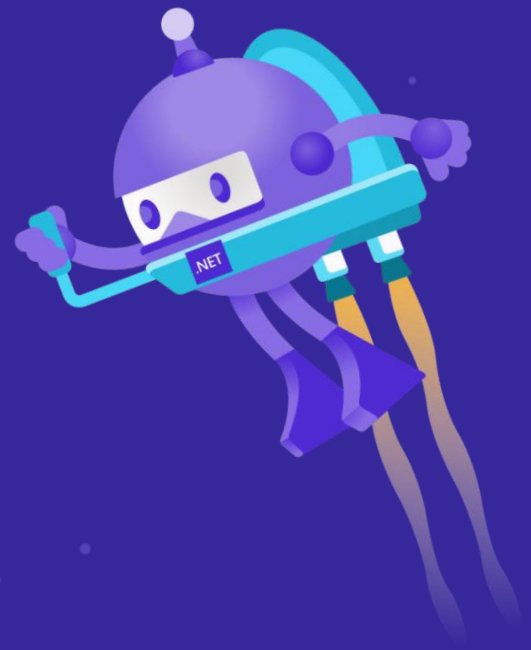
AddAuthenticationStateSerialization

- Adds necessary services to serialize the authentication state on the server

AddAuthenticationStateDeserialization

- Adds necessary services to deserialize the authentication state in the browser
- By default, only serializes the server-side name and role claims for access in the browser
- Option to include all claims

Demo



Constructor injection

- Razor components support constructor injection.

```
public partial class MyCustomComponent(NavigationManager navigation)
{
    private void HandleClick()
    {
        navigation.NavigateTo("/counter");
    }
}
```

Authentication

Pushed Authorization Requests

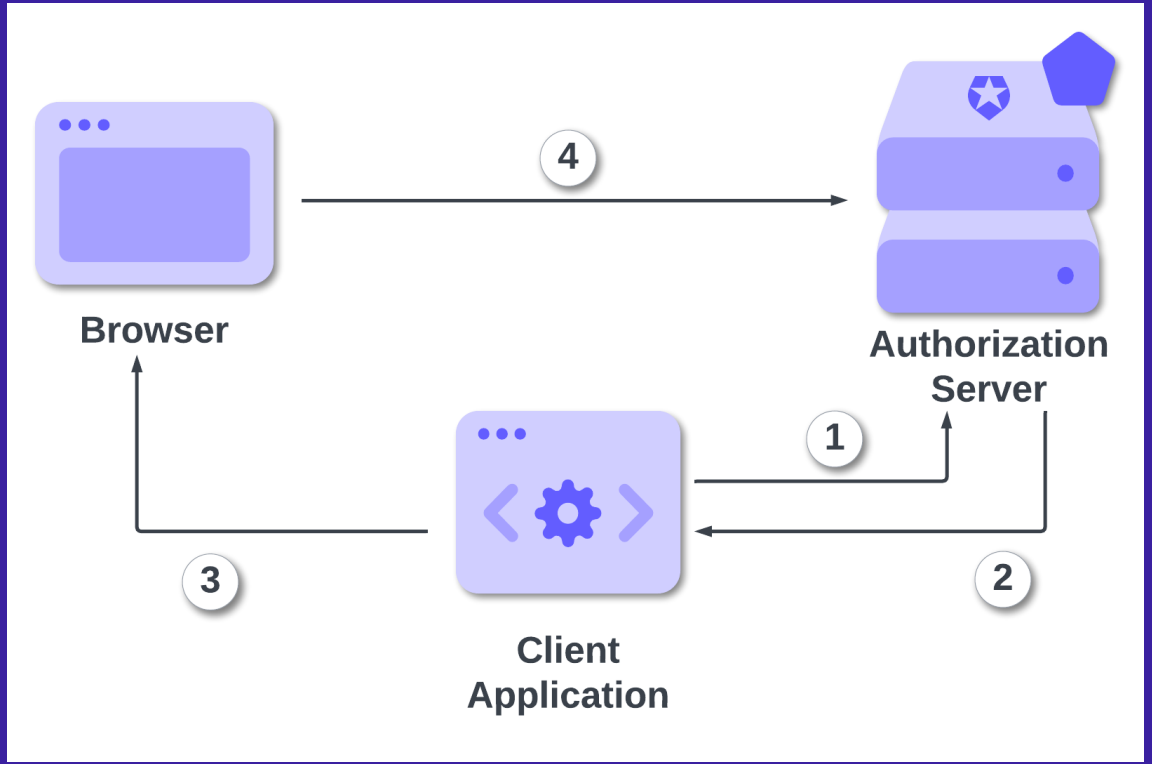
- **Pushed Authorization Requests: la nuova Frontiera dell'auth in .NET 9**
- Antonio Liccardi Antonio Venditti
- Room 1 – 17:00

Support for Pushed Authorization Requests (PAR)

- OAuth 2.0 and OpenID Connect extension that strengthens the security profile of these protocols
  - https://oauth.net/2/pushed-authorization-requests/
- PAR allows applications to not expose the parameters of an authorization request through the browser. Instead, it enables direct communication of these parameters between the application and the authorization server.

How PAR Works

- Instead of redirecting the user's browser to make the authorization request with all the parameters
- (1) the client application pushes that request directly to the authorization server with a POST request and (2) gets an identifier for that request.
- (3) Now, the client application redirects the user's browser to the authorization server with just that identifier to make its authorization request (4), avoiding exposing all the details.



HybridCache

HybridCache

- **HybridCache: la novità del caching in .NET9**
- Fabio Spaziani -> Room 2 - 16:10

New HybridCache library

- A new caching solution that combines the benefits of in-memory and distributed caching.
- In-Memory Caching:
  - Provides fast access to frequently used data.
  - Reduces latency by storing data in memory.
- Distributed Caching:
  - Ensures data consistency across multiple instances.
  - Supports scalability by distributing the cache across a cluster.
- Automatic Synchronization
  - Automatically synchronizes data between in-memory and distributed caches.
  - Ensures data consistency and reliability.

ASP.NET Core in .NET 9

Optimized static web asset handling

Improvements to exception handling and debugging

Authentication enhancements

New Blazor Hybrid Templates

Improved Kestrel connection metrics

SignalR improved distributed tracing

Dictionary debugging improvements

Built-in OpenAPI support

Detect Blazor component render mode

Improvements to DataProtection

SignalR AOT support

Blazor reconnection improvements

Trust Developer certs on Linux

Up to 25% faster Blazor startup

Keyed service support in middleware

Whats new in ASP.NET Core 10 (preview)

- Kestrel
  - Allow server Memory Pool to shrink #27394
- Authentication and Identity
  - Passkeys Authentication support in ASP.NET Core #53467
  - OAuth 2 refresh token support #8175
  - Refreshing auth tokens for SignalR #5297
  - Add Microsoft Identity Platform auth option to the Blazor Web App template #51202
  - Add ability to scaffold identity endpoints in project and override these #53343

Whats new in ASP.NET Core 10 (preview)

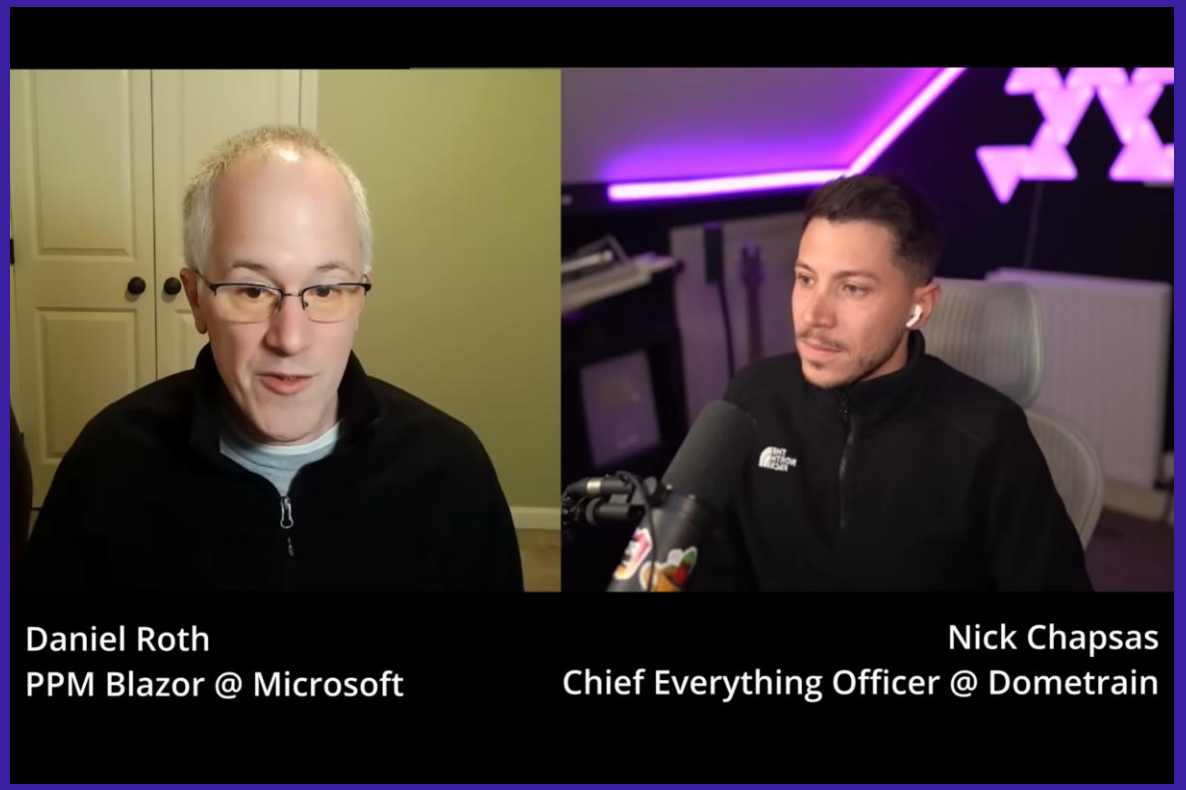
- Web UI
  - Blazor Server state persistence
    - [SupplyParameterFromPersistentComponentState]
    - [Blazor] Persisting circuit state for Blazor applications #60494
    - Declarative model to serialize state on prerendering with Blazor and restore it on the client #26794
    - APIs for components to persist state across circuits manually #30344
    - Blazor Server: Provide APIs/extensibility for circuit eviction #17866
    - Support persistent component state across enhanced page navigations #51584
  - Startup performance
    - Stop embedding JS files in Endpoints and Server assemblies #58783
    - Preload Blazor WebAssembly resources to improve startup time #58875
    - [Blazor] Get rid of Blazor.boot.json and use a different strategy to flow the environment. #59456

Whats new in ASP.NET Core 10 (preview)

- APIs
  - Support unified model for DataAnnotations-based validation via source generator #46349
  - JsonPatchDocument should use System.Text.Json #24333
  - Add support for emitting ServerSentEvents from minimal APIs #56172
- OpenAPI
  - Support XML-based OpenAPI docs for minimal and controller-based APIs with Microsoft.AspNetCore.OpenApi #39927
  - Support emitting OpenAPI documents in YAML format #58516
  - Upgrade Microsoft.AspNetCore.OpenAPI and add support for OpenAPI v3.1 #58619

Nick Chapsas
"I Confronted Microsoft About Blazor's Future"

- https://youtu.be/2uLGXe95kTo?si=WTWMCxLNxOi9LQg7



Questions?

```
{
    name:       Andrea Dottor
    email:      andrea@dottor.net
    bsky:       @andrea.dottor.net
    linkedin:   @andreadottor
}
```

.NET IN PILLOLE il podcast

Microsoft® MVP Most Valuable Professional

Questions?

- Vote for my session -