



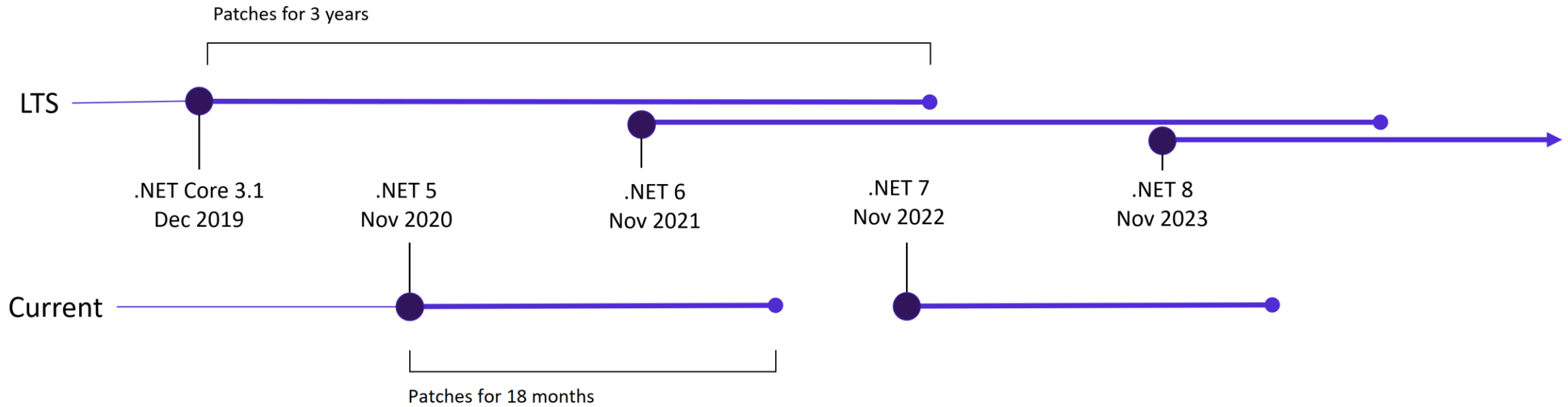
@RobertHaken 

Software & Cloud Architect | Microsoft MVP: Development | HAVIT, s.r.o.

# Novinky [ASP].NET 7, Blazor a C# 11



# .NET Release Cycle



# C# 11



C# 11

DEMO



# C# 11

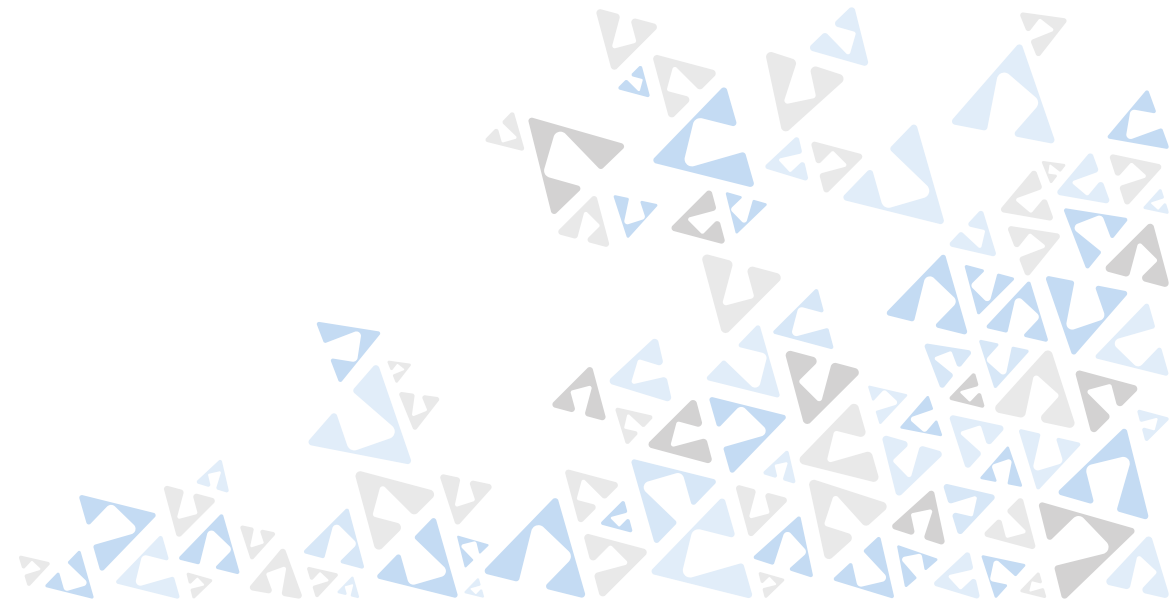
Required Members

Generic Attributes

Raw String Literals

List Patterns (+ Slice Pattern)

Generic Math Interfaces



# C# 11

Numeric IntPtr (`nint`) and UIntPtr (`nuint`)

Pattern match `Span<char>` on a constant `string`

UTF-8 string literals - `"AUTH"u8.ToArray()`

File local types

Generic Math

- static abstract members in interfaces
- checked user defined operators
- relaxed shift operators
- unsigned right-shift operator



# ASP.NET Core v .NET 7



# Novinky v Minimal APIs

OpenAPI (Swagger)

Endpoint Filters, Route Groups (MapGroup)

Streams, arrays, StringValues bindings

Anti-forgery support (anti-scrf)





# Output Caching

```
builder.Services.AddOutputCache([defaults & policies]);
```

```
app.UseOutputCache();
```

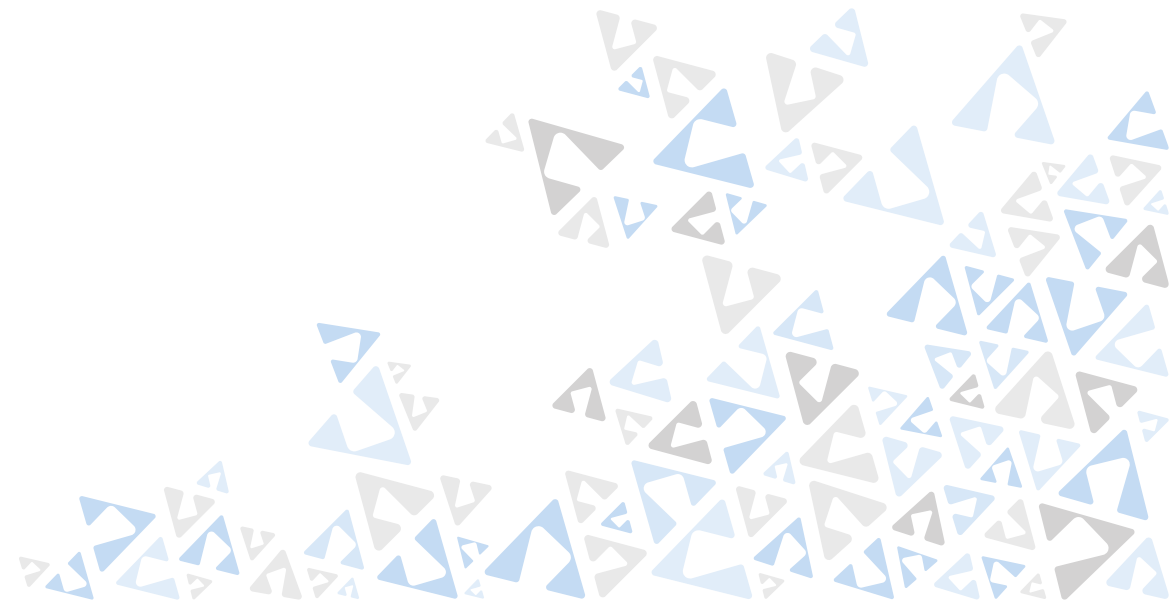
```
app.MapXy(...).CacheOutput(...);
```

```
[OutputCache]
```



Output Caching

DEMO



# Output Caching vs. Response Caching

## Output Caching

- driven by configuration
- stores: mem, disk, REDIS, Blob storage, ...
- invalidation with tags
- resource locking
- re-validation from clients (304 Not Modified)
- new `ResponseCachingPolicy()`

## Response Caching

- driven by HTTP headers
- store: mem



# Rate Limiter

Microsoft.AspNetCore.RateLimiting NuGet package

- RateLimiting middleware

```
app.UseRateLimiter(...options with policies...);  
[RateLimit("policyName")]  
.MapXy().RequireRateLimiting(policyName)
```

System.Threading.RateLimiting NuGet package

- standalone RateLimiter for your own code



# Rate Limiter II.

## Algorithms

- Concurrency limit
- Token bucket limit
- Fixed window limit
- Sliding window limit (segments)

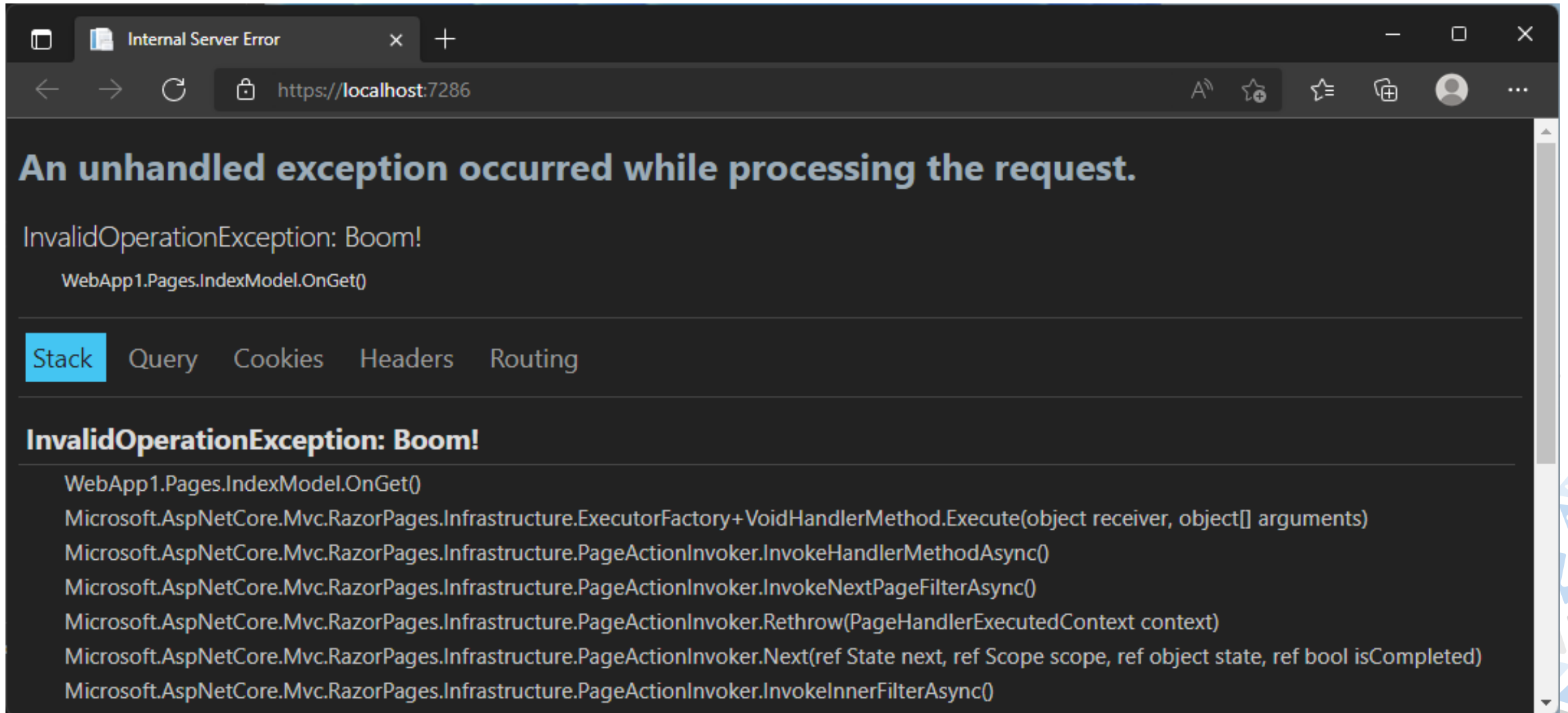


Rate Limiting

DEMO



# Developer exception page dark mode



Internal Server Error

https://localhost:7286

## An unhandled exception occurred while processing the request.

InvalidOperationException: Boom!  
WebApp1.Pages.IndexModel.OnGet()

Stack Query Cookies Headers Routing

### InvalidOperationException: Boom!

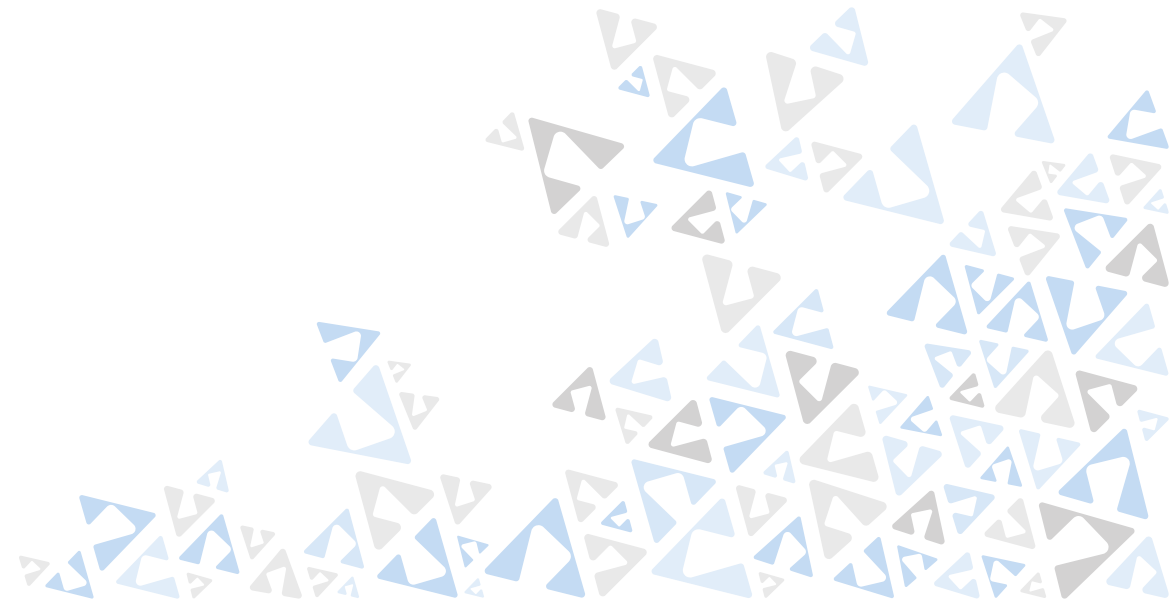
- WebApp1.Pages.IndexModel.OnGet()
- Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.ExecutorFactory+VoidHandlerMethod.Execute(object receiver, object[] arguments)
- Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.PageActionInvoker.InvokeHandlerMethodAsync()
- Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.PageActionInvoker.InvokeNextPageFilterAsync()
- Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.PageActionInvoker.Rethrow(PageHandlerExecutedContext context)
- Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.PageActionInvoker.Next(ref State next, ref Scope scope, ref object state, ref bool isCompleted)
- Microsoft.AspNetCore.Mvc.RazorPages.Infrastructure.PageActionInvoker.InvokeInnerFilterAsync()

# gRPC JSON Transcoding

RESTful HTTP APIs for gRPC services

call gRPC methods with familiar HTTP concepts

- HTTP verbs
- JSON requests/responses
- URL parameter binding





# ...and more

[HTTP/3 \(preview in net6\)](#)

[Bind using TryParse in MVC and API Controllers](#)

[Infer API controller action parameters that come from services](#)

[New problem details service](#)

[Request decompression middleware](#)

[New ServerReady event for measuring startup time](#)

[SignalR: Support returning values from client invocation](#)



# Blazor v .NET 7



# Blazor Hybrid

MAUI

WPF

WinForms

BlazorWebView control





# Blazor Hybrid as team top-priority



- Expose location changing event for NavigationManger #14962
- Calling InvokeAsync(StateHasChanged) causes page to fallback to default culture #28521
- Empty Blazor project template #32832
- Productize the custom elements support package #38447
- Improve prerendering and authentication experience in Blazor #27592
- Design proposal: Bind get/set/after modifiers #39837
- Enable integration with 3rd party assets management solutions #38445
- Components don't reset their received parameters during hot reload updates #31272
- [Blazor][Wasm] Dynamic and extensible authentication requests #42580
- Better MSAL support for additional scenarios #38122
- Developers targeting browser-wasm can use Web Crypto APIs runtime#40074
- Continued improvements to Mono Method Body Replacement runtime#57365
- Add DateOnly and TimeOnly support to model binding & routing #34591
- Blazor Image component to display images that are not accessible through HTTP endpoints #25274
- Real multithreading (on supported browsers) #17730
- Ability to run multiple Blazor server / Web assembly apps in the same document #38128
- Perform a pass cleaning up the web templates #30162
- Consider shipping ObjectGraphDataAnnotationsValidator / ValidateComplexTypeAttribute #28640
- Design for solving transient disposables on Blazor Server #26676
- Pause and resume Blazor Server circuits #30344
- How to configure HttpClient base address in Blazor Server using IHttpConnectionFactory #25758
- Ability to monitor circuit activity #30287
- Blazor Server: Provide APIs/extensibility for circuit eviction #17866



# Blazor data binding **get/set/after** modifiers

**@bind:after** modifier runs async logic after a binding event has completed

```
<input @bind="searchText" @bind:after="PerformSearch" />
```

```
@code {  
    string searchText;  
  
    async Task PerformSearch()  
    {  
        // ... do something asynchronously with 'searchText' ...  
    }  
}
```



@bind:after

DEMO



# Blazor data binding **get/set/after** modifiers II.

**@bind:set** and **@bind:get** (always together) for more binding control

```
<input @bind:get="Value" @bind:set="HandleValueChanged" />
```

```
@code {  
    [Parameter] public TValue Value { get; set; }  
    [Parameter] public EventCallback<string> ValueChanged { get; set; }  
    private Task HandleValueChanged(string newValue)  
    {  
        Value = newValue;  
        await ValueChanged.InvokeAsync(Value);  
    }  
}
```



# Passing state using NavigationManager

store state when navigating

```
NavigationManager.NavigateTo("page", new NavigationOptions() { HistoryEntryState = "some state" });
```

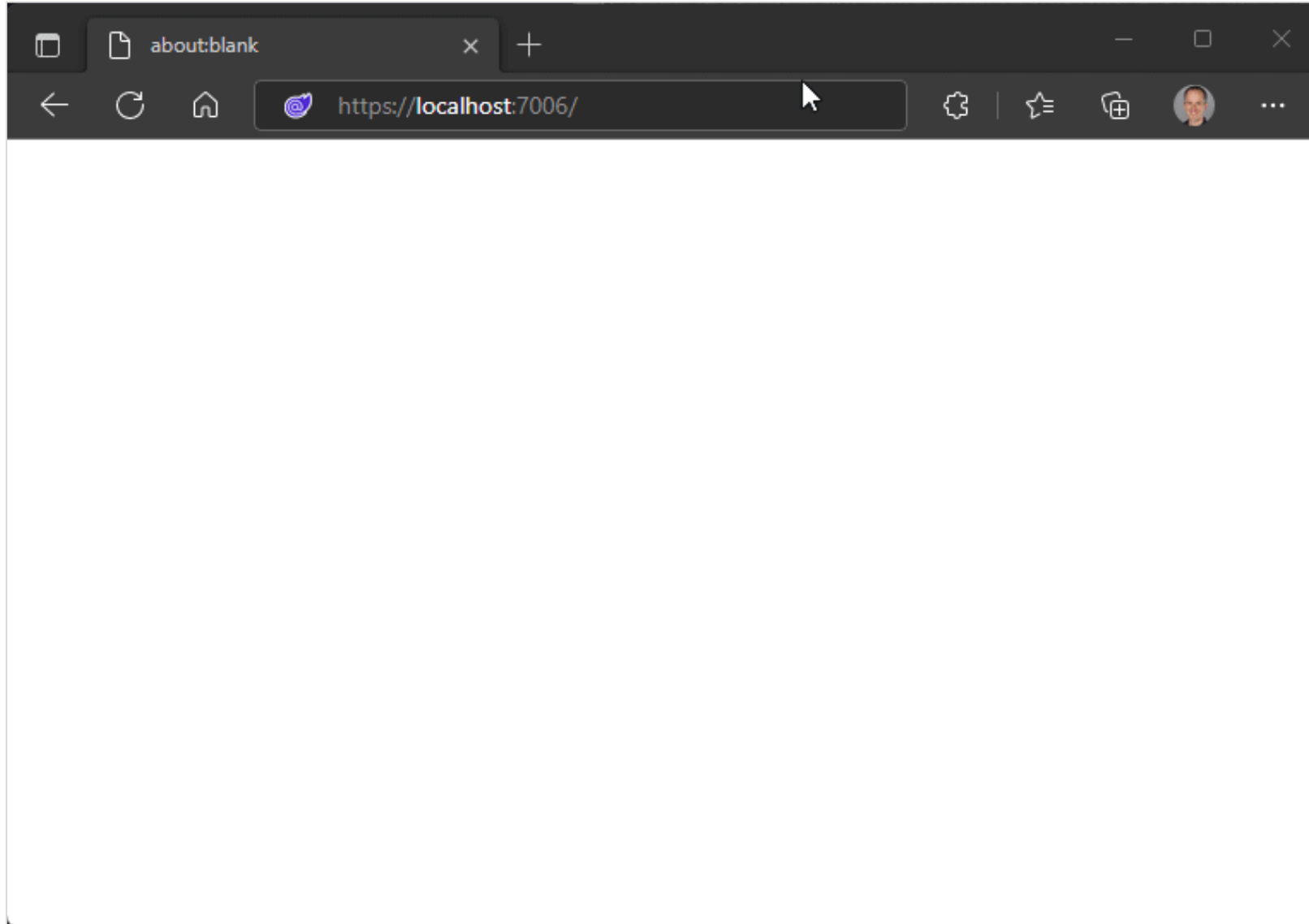
pickup state from navigation

```
string state = NavigationManager.HistoryEntryState;
```





# Blazor WebAssembly loading page



New CSS variables provided

`--blazor-load-percentage`

`--blazor-load-percentage-text`



# Cancellation of navigation events

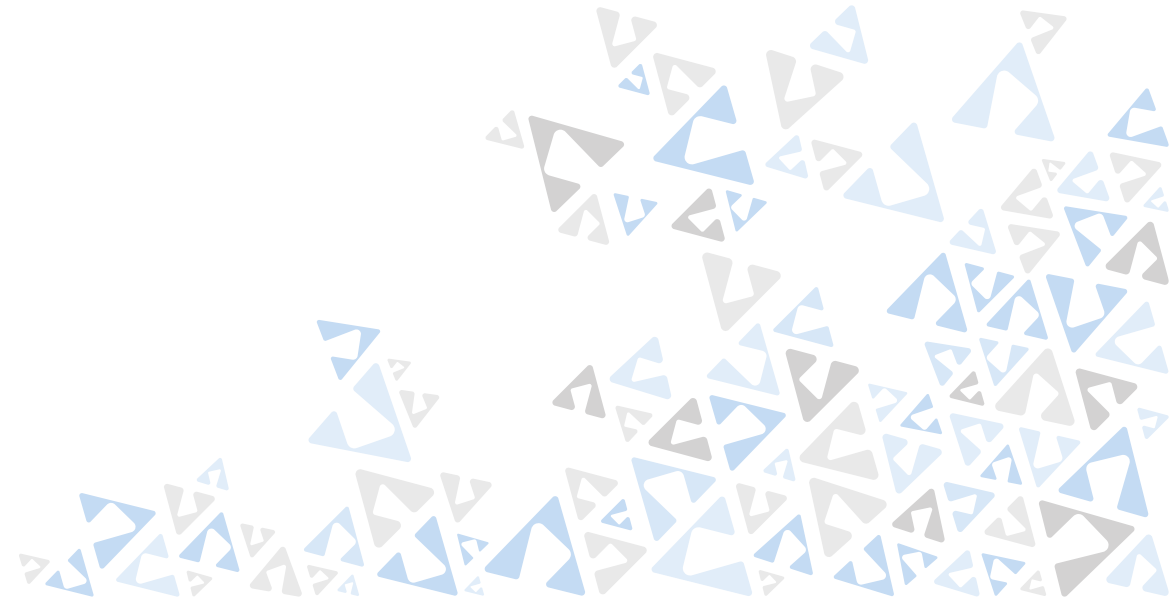
```
@if (_isDirty)
{
    <NavigationLock OnBeforeInternalNavigation="OnBeforeInternalNavigation"
        ConfirmExternalNavigation="true" />
}
@code {
    private async Task OnBeforeInternalNavigation(LocationChangingContext context)
    {
        var isConfirmed = await MessageBox.Confirm("Are you sure you want to continue?");
        if (!isConfirmed)
        {
            context.PreventNavigation();
        }
    }
}
```

```
NavigationManager.RegisterLocationChangingHandler(OnLocationChanging);
```

# Blazor Empty Project Template

demo code removed

Bootstrap removed



# Blazor **Virtualize** spacer customization

default spacer is **div**

NET7 allows you to customize the spacer element

```
<tbody>  
  <Virtualize SpacerElement="tr">  
    ...  
  </Virtualize>  
</tbody>
```



# ...and more

improved performance, hot-reloads, debugging

[Inject services into custom validation attributes in Blazor](#) (ValidationContext as service provider)

[Custom elements \(preview in net6\)](#)

[Prerendering with authentication](#)

[Additional System.Security.Cryptography support on WebAssembly](#)

[Better MSAL support, additional scenarios](#)

Mixed mode AOT?

.NET WebAssembly without Blazor UI (preview)

[Multiple Blazor Apps in same document](#)

[ObjectGraphDataAnnotationsValidator / ValidateComplexTypeAttribute](#)

[Real multithreading](#)



# Entity Framework [Core] 7



# Entity Framework [Core] 7

JSON Columns – mapping, queries, updates

ExecuteUpdate() and ExecuteDelete() bulk operations

Faster SaveChanges() – eliminated transactions, ...

Table-per-concrete-type (TPC) inheritance mapping

Custom Reverse Engineering Templates (DB first)

Model building conventions

Stored procedure mapping (insert, update, delete)

New and improved interceptors and events

...



# Resources

Dema

<https://github.com/hakenr/CSharp11Demo>

<https://github.com/hakenr/AspNetAndBlazorNet7Demos>

Havit.Blazor komponenty pro Bootstrap

<https://havit.blazor.eu/>

HAVIT YouTube channel

<https://www.youtube.com/user/HAVITcz>

