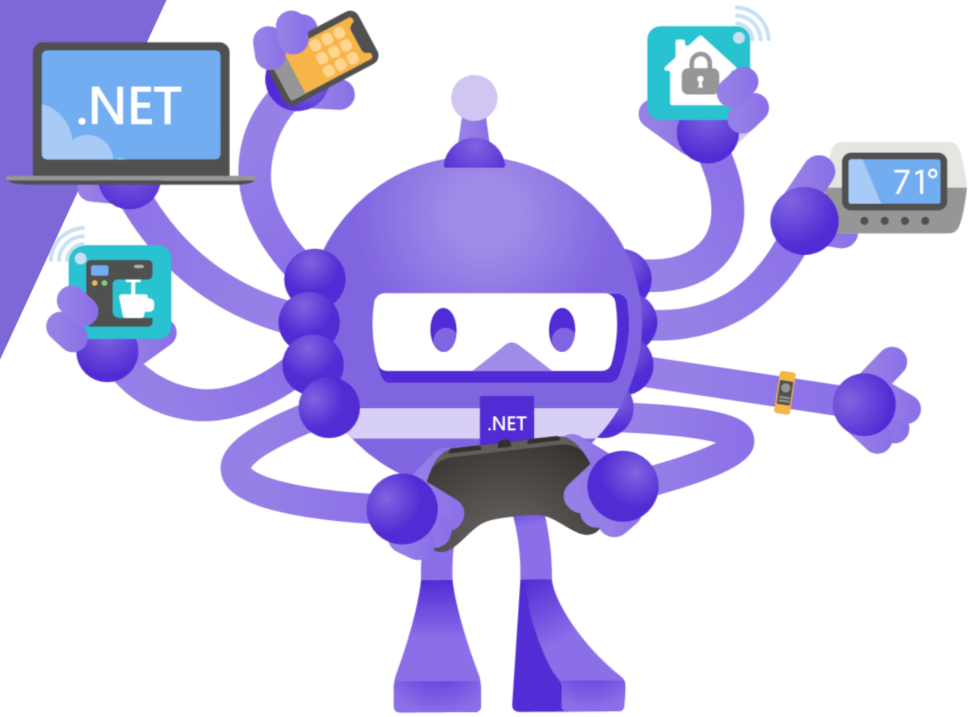
.NET Conference 2023

Come migliorare la nostra applicazione web con .NET 7

Andrea Dottor
@dottor



SPONSOR

Microsoft

avanade

altitudo

BC Soft
consulenza sistemi informativi

Reti
Società Benefit

UNIKEY
Bringing IT knowledge to the people

JET BRAINS

Packt>

NET CODE

NET CODE

Image by tookapic from Pixabay

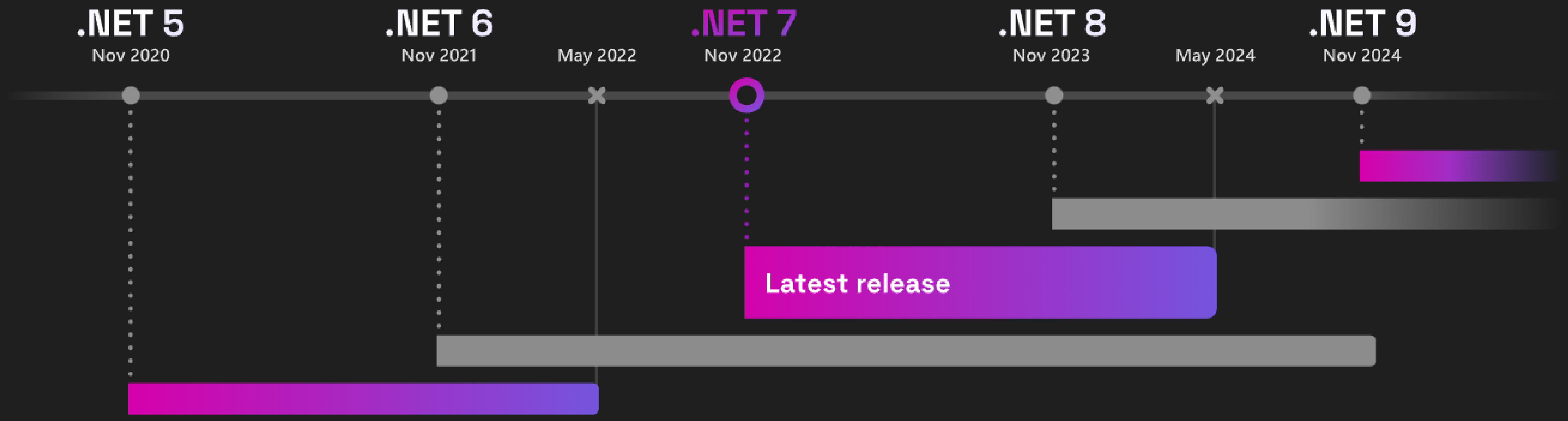
Codice e slide

BrewerApp
https://github.com/andreadottor/BrewerApp

Why stay on .NET 6?

> .NET 7 is not an LTS.

LTS vs STS



🟪 **STANDARD TERM SUPPORT**
Patches for 18 months

⬜ **LONG TERM SUPPORT**
Patches for 36 months

LTS vs STS

Long Term Support (LTS) releases get free support and patches for 3 years

- .NET 6 – End of support: November 12, 2024

Standard Term Support (STS) releases get free support and patches for 18 months.

- supported for six months after a subsequent STS or LTS
- .NET 7 – End of support: May 14, 2024

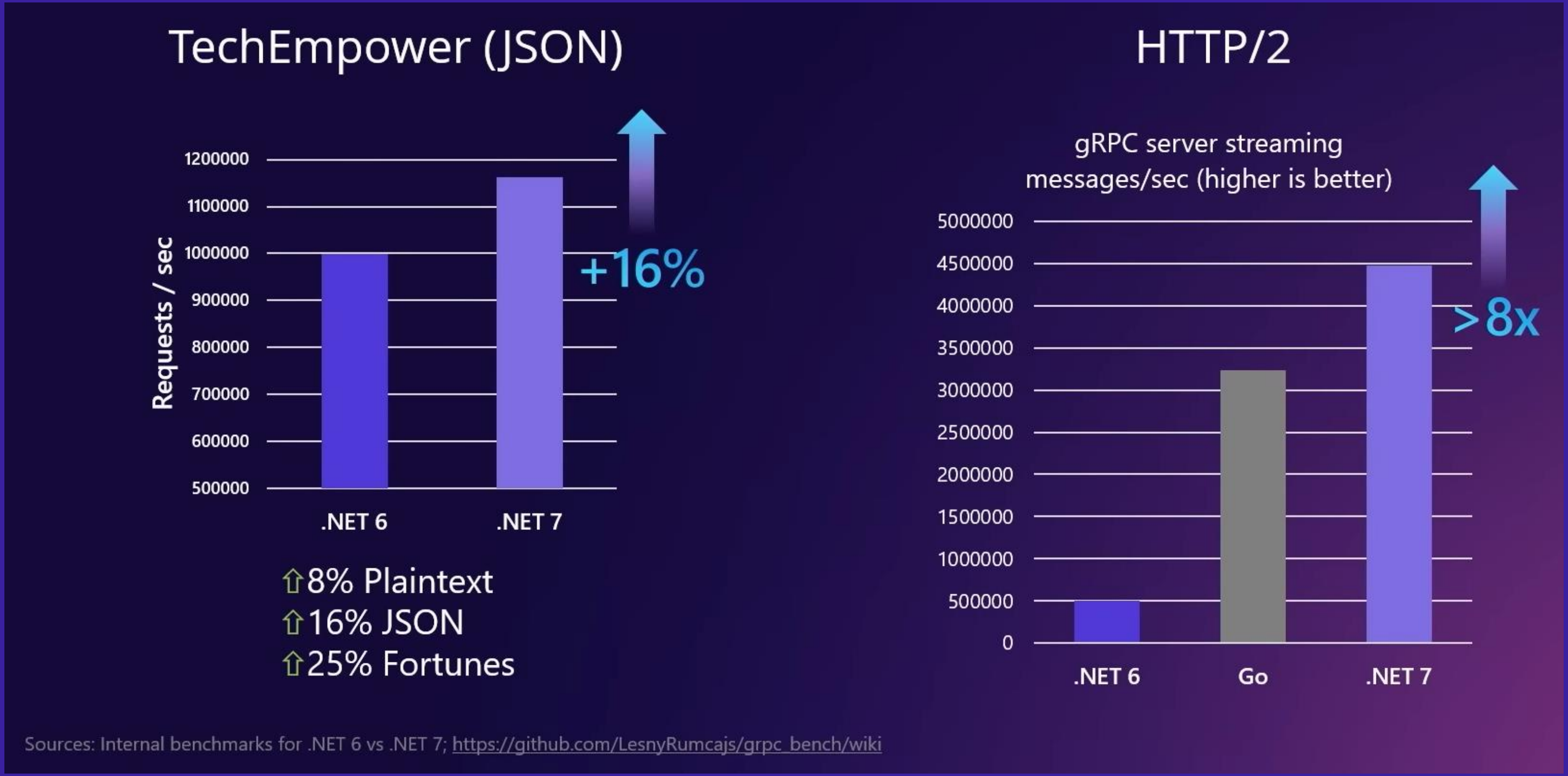
Updates are released on the Microsoft "Patch Tuesday" (second Tuesday of each month), however there is no guarantee that there will be a .NET release on any given Patch Tuesday

Why upgrade to .NET 7?

> Performance

> New features

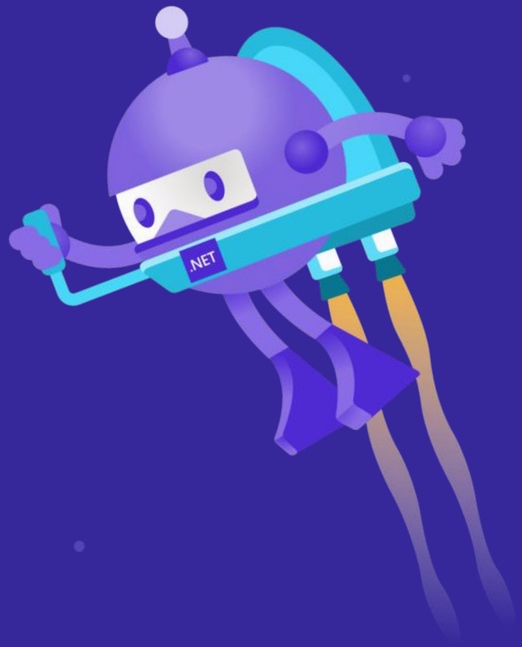
Performance



Sources: Internal benchmarks for .NET 6 vs .NET 7; https://github.com/LesnyRumcajs/grpc_bench/wiki

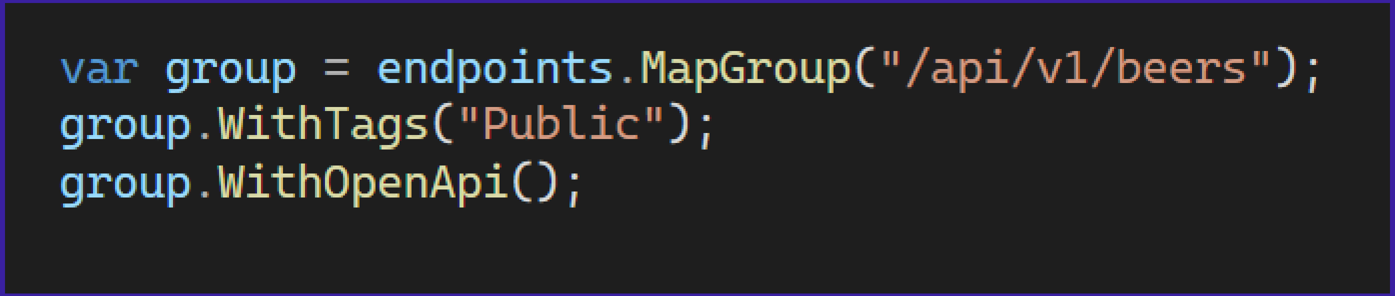
- https://devblogs.microsoft.com/dotnet/performance-improvements-in-aspnet-core-7/
- https://github.com/aspnet/Benchmarks
- (PowerBI) https://aka.ms/aspnet/benchmarks

Minimal API



Minimal API – Route groups

The **MapGroup** extension method helps organize groups of endpoints with a common prefix. It reduces repetitive code and allows for customizing entire groups of endpoints with a single call to methods like RequireAuthorization and WithMetadata which add endpoint metadata.

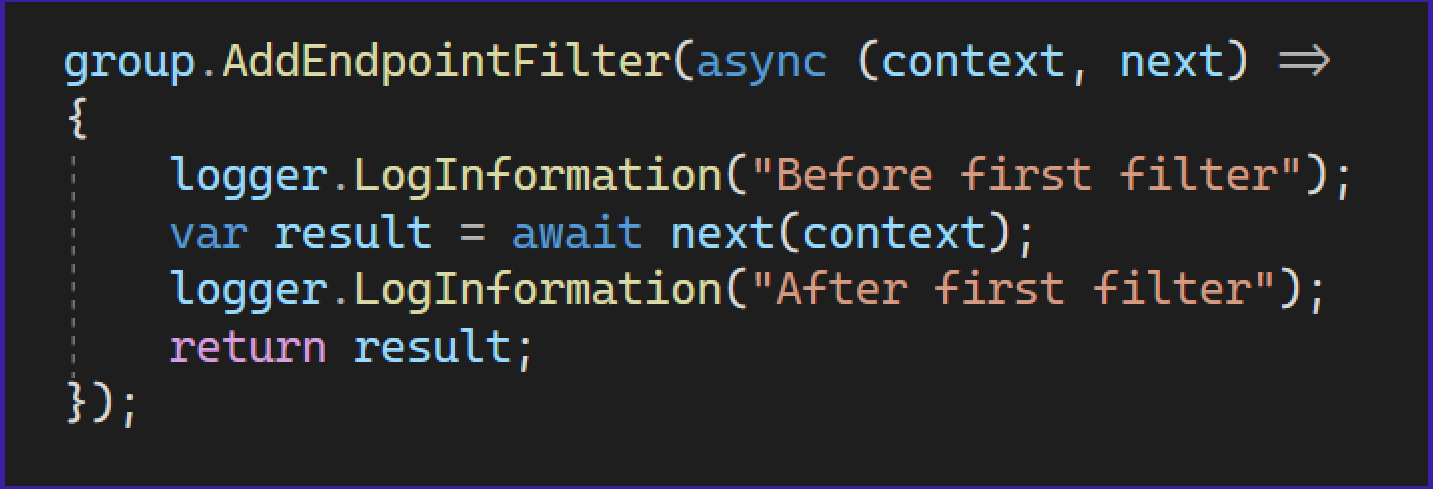


```
var group = endpoints.MapGroup("/api/v1/beers");
group.WithTags("Public");
group.WithOpenApi();
```

Filters in Minimal API

Allow developers to implement business logic that supports:

- Running code before and after the endpoint handler.
- Inspecting and modifying parameters provided during an endpoint handler invocation.
- Intercepting the response behavior of an endpoint handler.

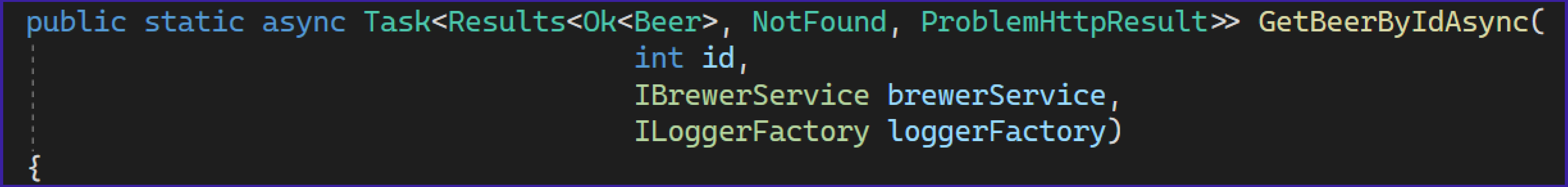


```csharp
group.AddEndpointFilter(async (context, next) =>
{
    logger.LogInformation("Before first filter");
    var result = await next(context);
    logger.LogInformation("After first filter");
    return result;
});
```

Return multiple result types

The new **Results<TResult1, TResult2, TResultN>** generic union types, along with the **TypesResults** class, can be used to declare that a route handler returns multiple **IResult**

- any of those types implementing IEndpointMetadataProvider will contribute to the endpoint’s metadata
- self-documenting API


```
public static async Task<Results<Ok<Beer>, NotFound, ProblemHttpResult>> GetBeerByIdAsync(
                                    int id,
                                    IBrewerService brewerService,
                                    ILoggerFactory loggerFactory)
{
```

Minimal API – OpenAPI improvements

The **Microsoft.AspNetCore.OpenApi** package allows interactions with OpenAPI specifications for endpoints.

Calling **WithOpenApi** on the endpoint adds to the endpoint's metadata.

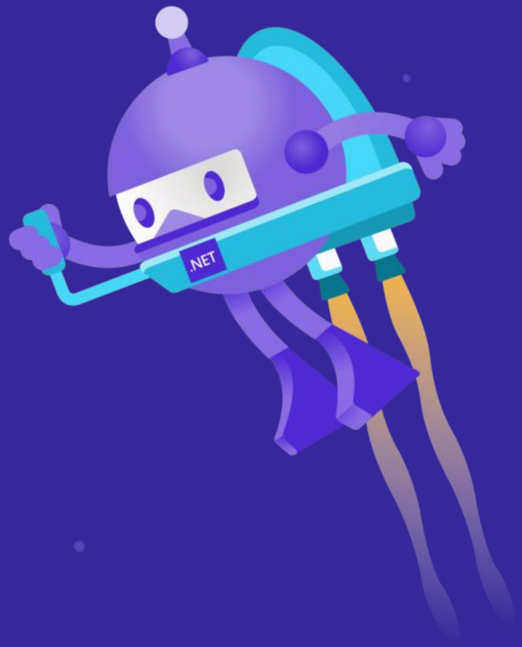
Now support annotating operations with descriptions and summaries: you can call extension methods **WithDescription** and **WithSummary.**

DEMO

Image by Alexander Lesnitsky from Pixabay

ASP.NET Core

Razor Pages, MVC, Controllers, Minimal API



Output caching middleware

Output caching is a new middleware that stores responses from a web app and serves them from a cache rather than computing them every time.

Can be used in all types of ASP.NET Core apps: Minimal API, Web API with controllers, MVC, and Razor Pages.

Default output caching policy (*can be override*)

- Only HTTP 200 responses are cached.
- Only HTTP GET or HEAD requests are cached.
- Responses that set cookies aren't cached.
- Responses to authenticated requests aren't cached.

https://learn.microsoft.com/en-us/aspnet/core/performance/caching/output?view=aspnetcore-7.0

Rate limiting

Rate limiting is the concept of limiting how much a resource can be accessed.

A way to control the amount of traffic that a web application or API receives, by limiting the number of requests that can be made in a given period of time.

Can help to improve the performance of the site or application, and to prevent it from becoming unresponsive.

Rate limiter algorithms

- Fixed window
- Sliding window
- Token bucket
- Concurrency

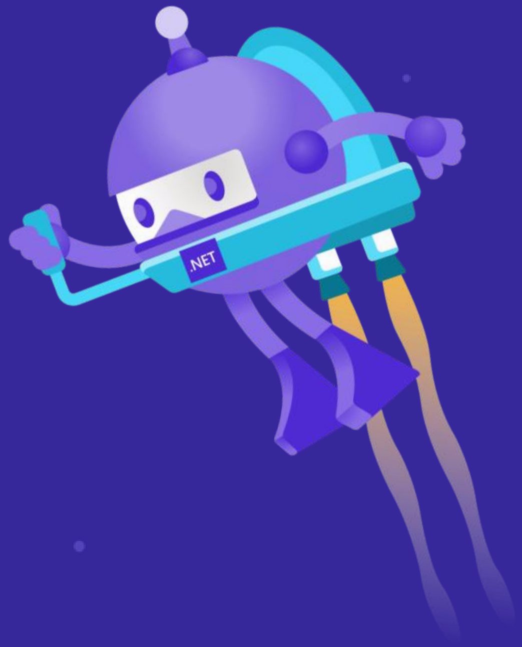
https://devblogs.microsoft.com/dotnet/announcing-rate-limiting-for-dotnet/

https://learn.microsoft.com/en-us/aspnet/core/performance/rate-limit?preserve-view=true&view=aspnetcore-7.0

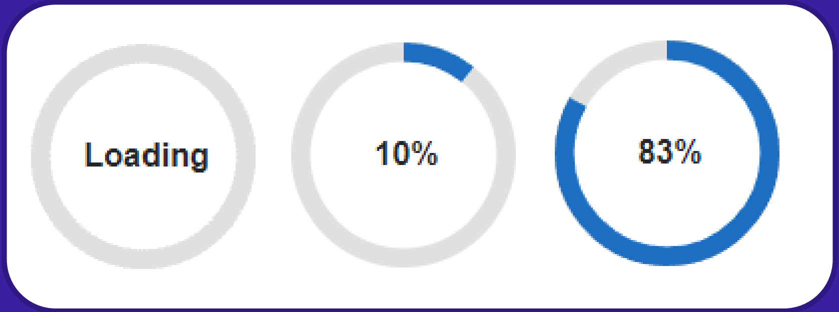
DEMO

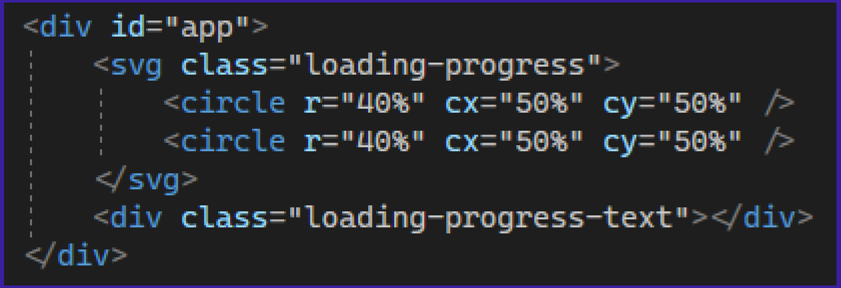
Image by Alexander Lesnitsky from Pixabay

Blazor

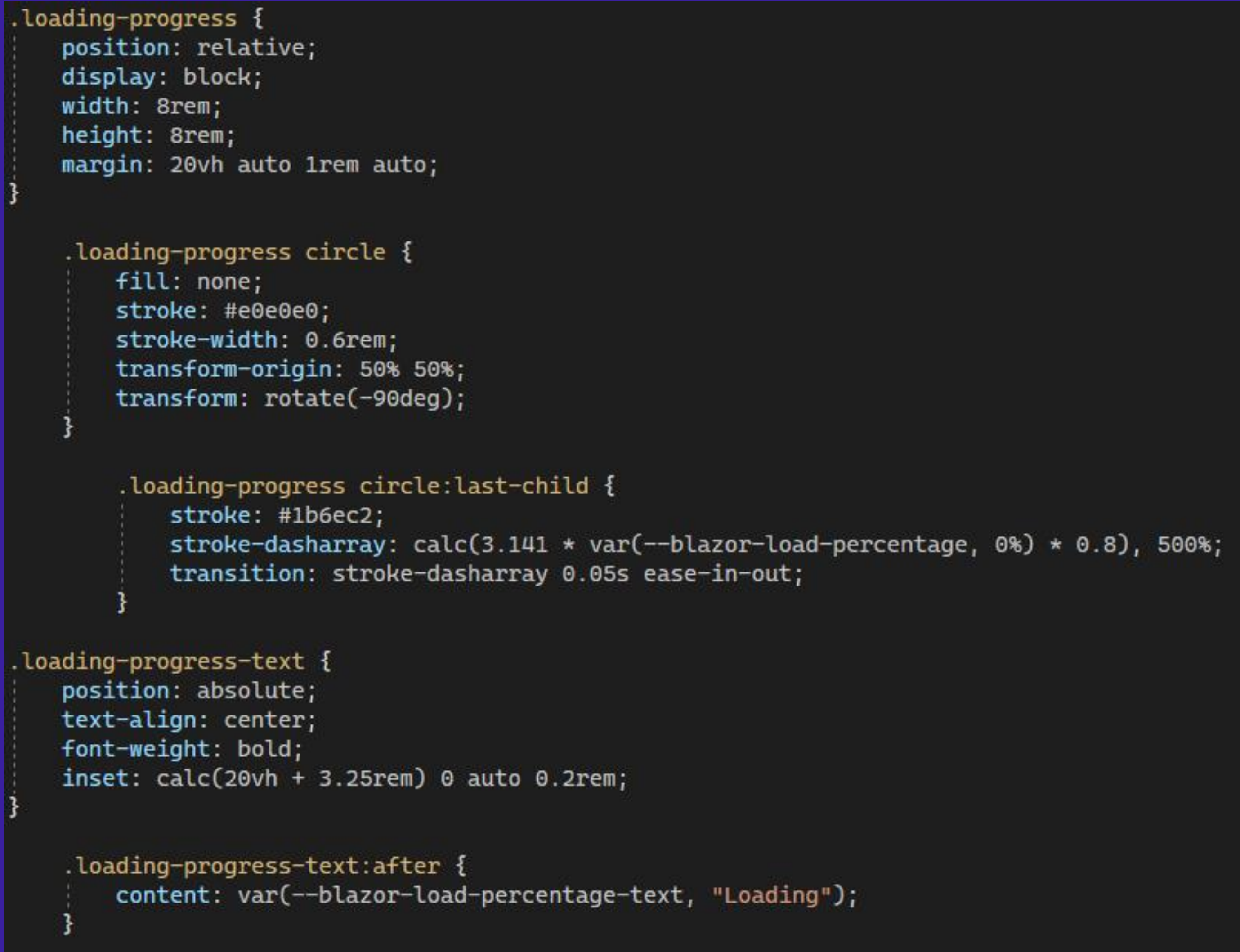


Loading progress indicators (only Blazor WebAssembly)





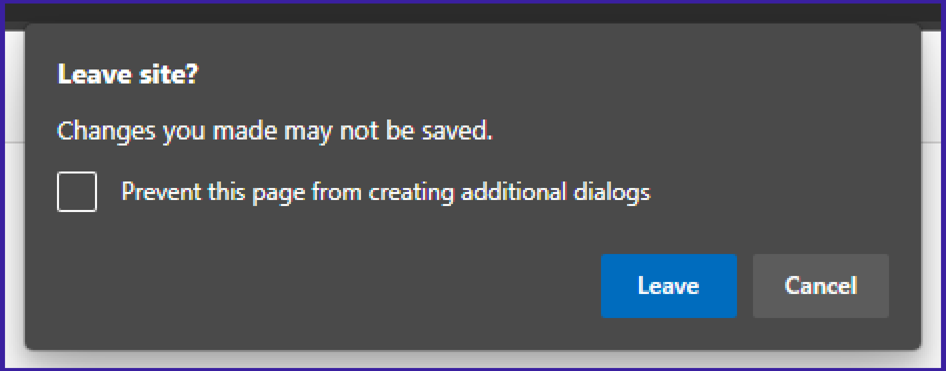
```html
<div id="app">
    <svg class="loading-progress">
        <circle r="40%" cx="50%" cy="50%" />
        <circle r="40%" cx="50%" cy="50%" />
    </svg>
    <div class="loading-progress-text"></div>
</div>
```



```css
.loading-progress {
    position: relative;
    display: block;
    width: 8rem;
    height: 8rem;
    margin: 20vh auto 1rem auto;
}

    .loading-progress circle {
        fill: none;
        stroke: #e0e0e0;
        stroke-width: 0.6rem;
        transform-origin: 50% 50%;
        transform: rotate(-90deg);
    }

        .loading-progress circle:last-child {
            stroke: #1b6ec2;
            stroke-dasharray: calc(3.141 * var(--blazor-load-percentage, 0%) * 0.8), 500%;
            transition: stroke-dasharray 0.05s ease-in-out;
        }

.loading-progress-text {
    position: absolute;
    text-align: center;
    font-weight: bold;
    inset: calc(20vh + 3.25rem) 0 auto 0.2rem;
}

    .loading-progress-text:after {
        content: var(--blazor-load-percentage-text, "Loading");
    }
```

Handle location changing events and navigation state

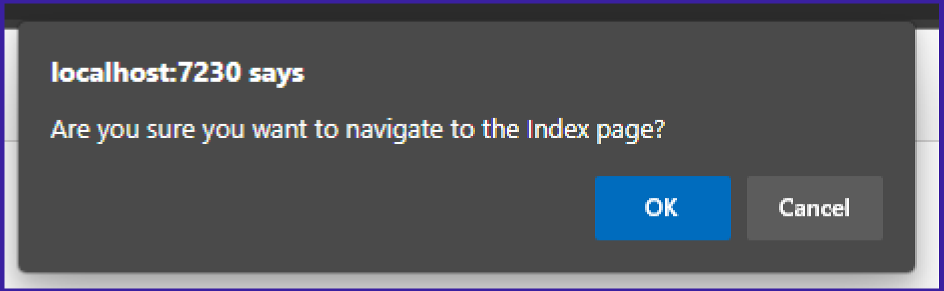
- NavigationLock component
- Navigation.RegisterLocationChangingHandler



Leave site?

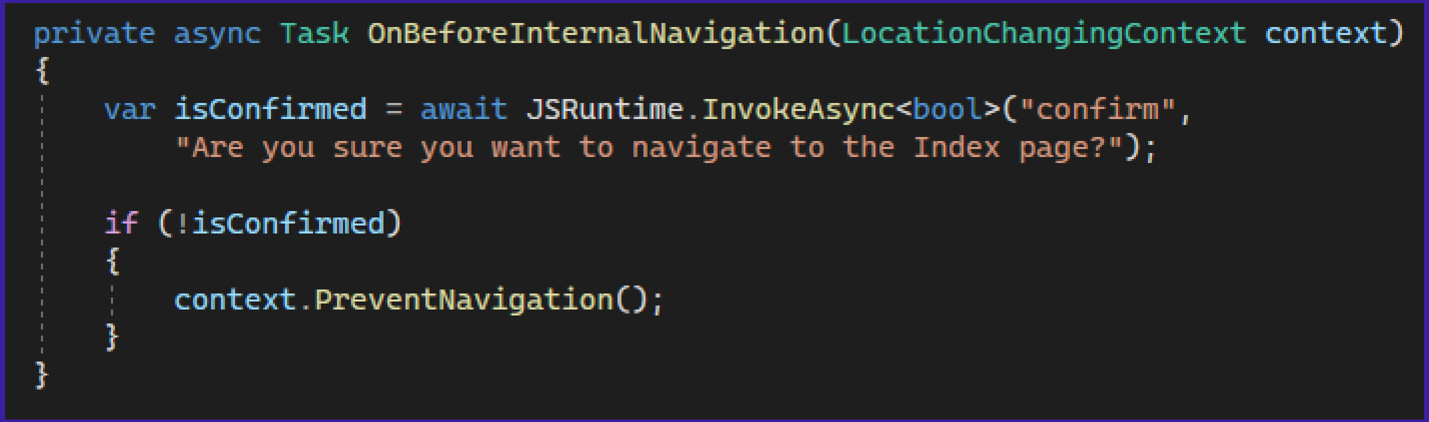
Changes you made may not be saved.

☐ Prevent this page from creating additional dialogs



localhost:7230 says

Are you sure you want to navigate to the Index page?



```
private async Task OnBeforeInternalNavigation(LocationChangingContext context)
{
    var isConfirmed = await JSRuntime.InvokeAsync<bool>("confirm",
        "Are you sure you want to navigate to the Index page?");

    if (!isConfirmed)
    {
        context.PreventNavigation();
    }
}
```

Blazor custom elements

Use Blazor custom elements to dynamically render Razor components from other SPA frameworks, such as Angular or React.

Blazor custom elements:

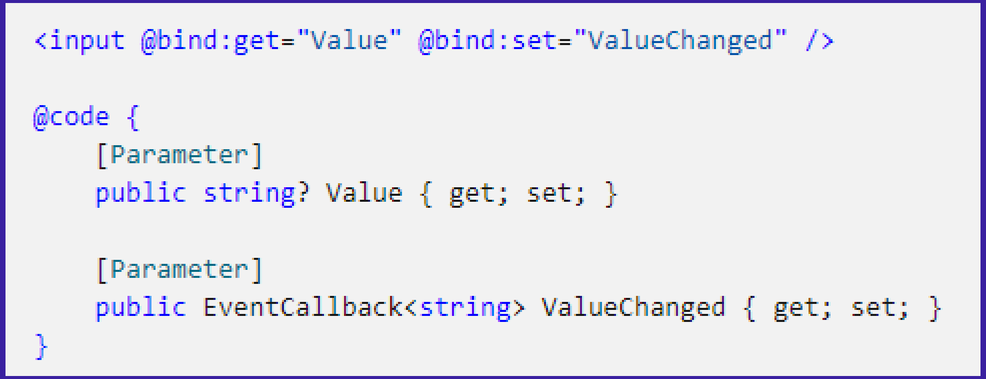
- Use standard HTML interfaces to implement custom HTML elements.
- Eliminate the need to manually manage the state and lifecycle of root Razor components using JavaScript APIs.
- Are useful for gradually introducing Razor components into existing projects written in other SPA frameworks.

DEMO

Image by Koltrein from Pixabay

Bind after/get/set modifiers

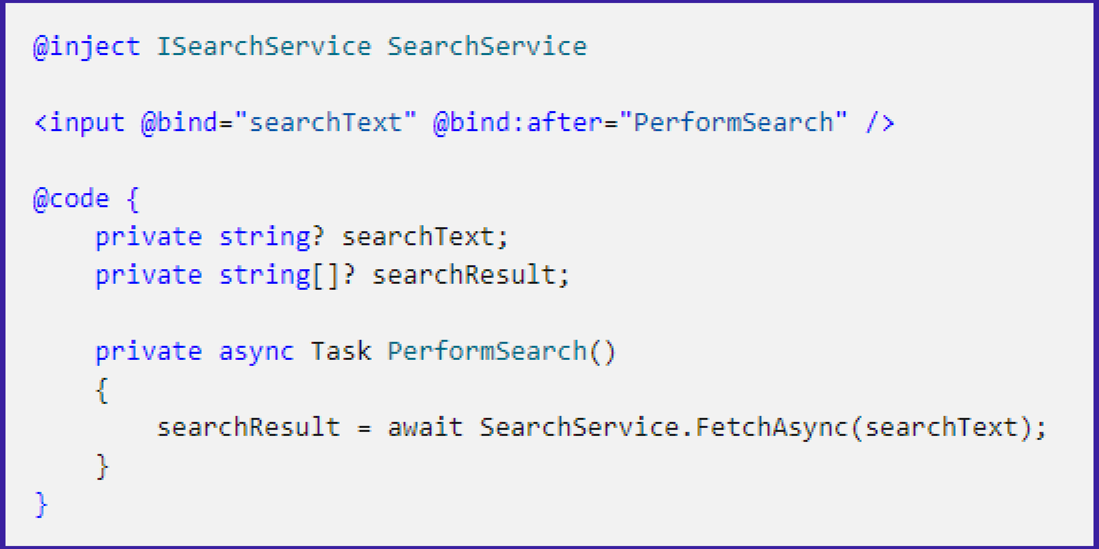
• @bind:get	Specifies the value to bind.
• @bind:set	Specifies a callback for when the value changes.



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

@code {
    [Parameter]
    public string? Value { get; set; }

    [Parameter]
    public EventCallback<string> ValueChanged { get; set; }
}
```

Bind after/get/set modifiers

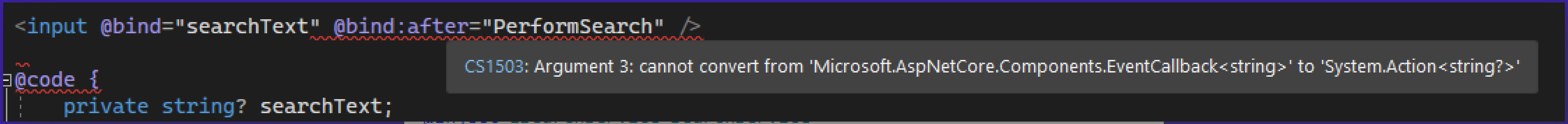
• @bind:get	Specifies the value to bind.
• @bind:set	Specifies a callback for when the value changes.
• @bind:after	To execute asynchronous logic after binding



```
@inject ISearchService SearchService

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

@code {
    private string? searchText;
    private string[]? searchResult;

    private async Task PerformSearch()
    {
        searchResult = await SearchService.FetchAsync(searchText);
    }
}
```

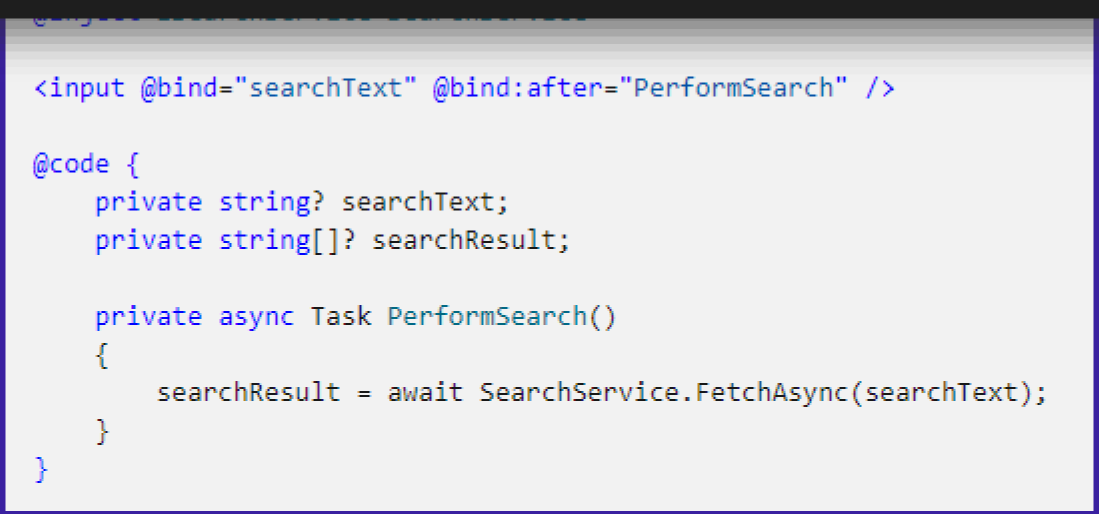
Bind after/get/set modifiers

• @bind:get	Specifies the value to bind.
• @bind:set	Specifies a callback for when the value changes.
• @bind:after	To execute asynchronous logic after binding



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

@code {
    private string? searchText;
```

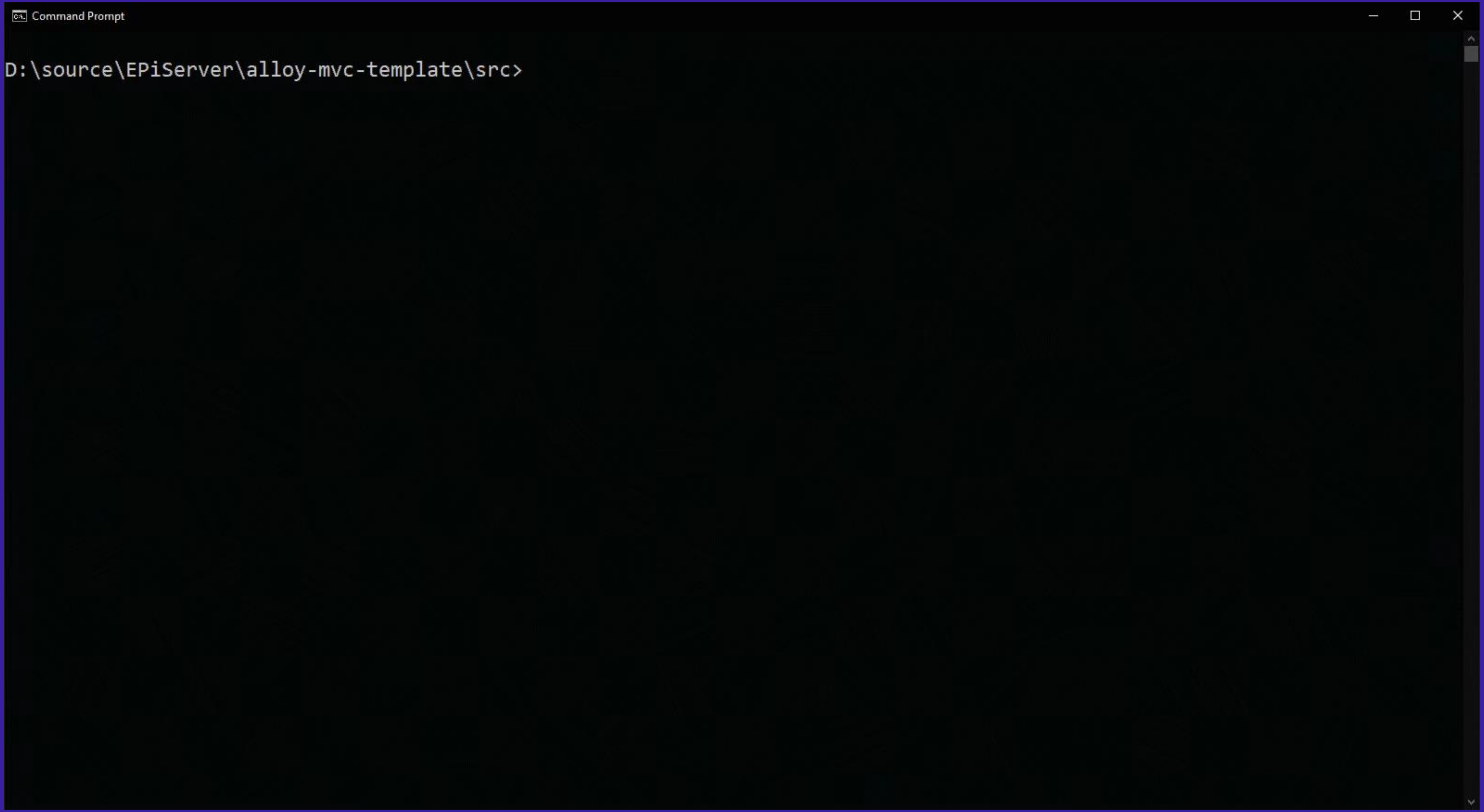
CS1503: Argument 3: cannot convert from 'Microsoft.AspNetCore.Components.EventCallback<string>' to 'System.Action<string?>'



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

@code {
    private string? searchText;
    private string[]? searchResult;

    private async Task PerformSearch()
    {
        searchResult = await SearchService.FetchAsync(searchText);
    }
}
```

Upgrade to .NET 7

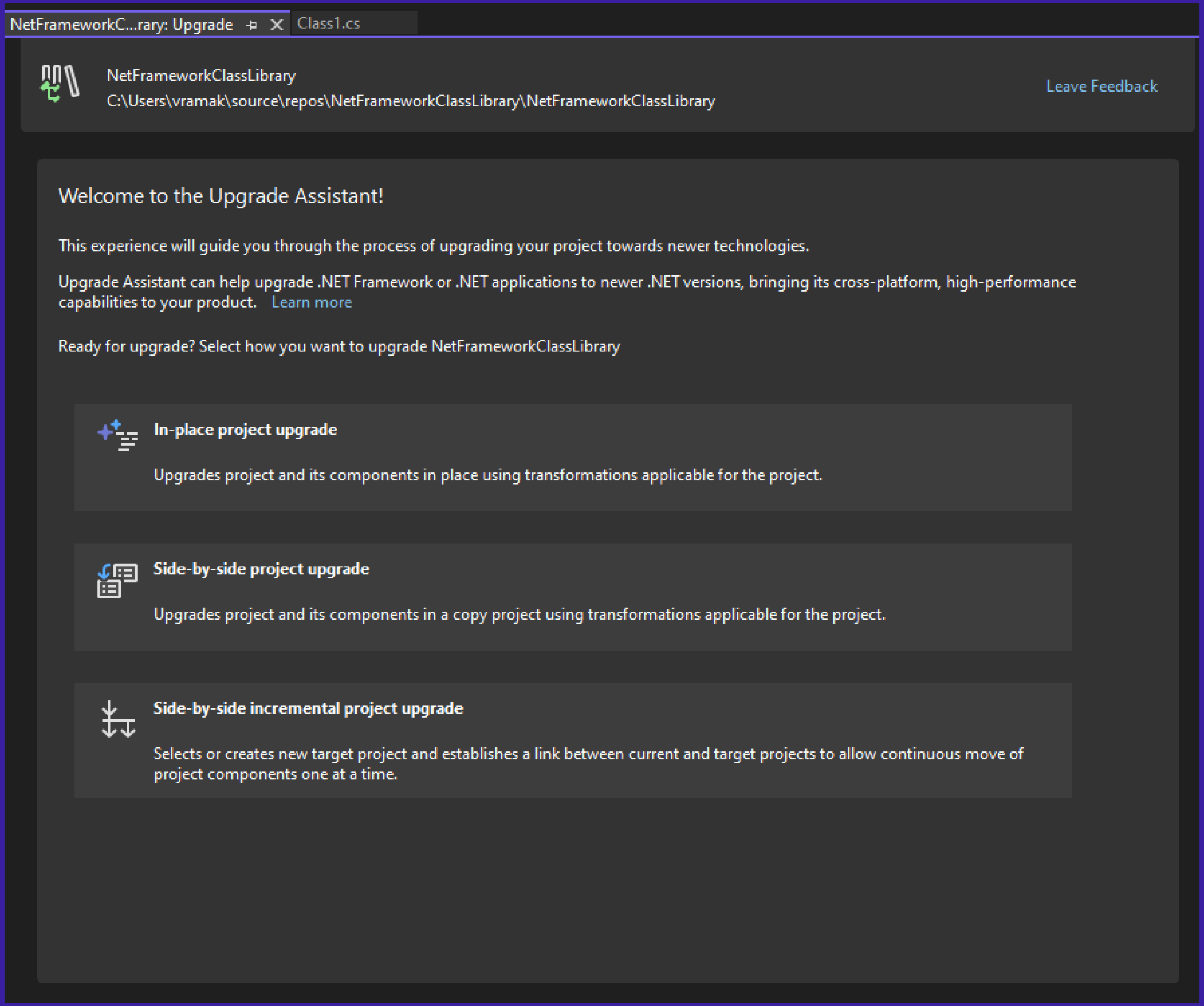
.NET Upgrade Assistant – dotnet tool



```
D:\source\EPiServer\alloy-mvc-template\src>
```

https://dotnet.microsoft.com/en-us/platform/upgrade-assistant/tutorial/intro

.NET Upgrade Assistant – VS extension



NetFrameworkClassLibrary
C:\Users\vramak\source\repos\NetFrameworkClassLibrary\NetFrameworkClassLibrary

Leave Feedback

Welcome to the Upgrade Assistant!

This experience will guide you through the process of upgrading your project towards newer technologies.

Upgrade Assistant can help upgrade .NET Framework or .NET applications to newer .NET versions, bringing its cross-platform, high-performance capabilities to your product. Learn more

Ready for upgrade? Select how you want to upgrade NetFrameworkClassLibrary

In-place project upgrade

Upgrades project and its components in place using transformations applicable for the project.

Side-by-side project upgrade

Upgrades project and its components in a copy project using transformations applicable for the project.

Side-by-side incremental project upgrade

Selects or creates new target project and establishes a link between current and target projects to allow continuous move of project components one at a time.

https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.upgradeassistant

Questions?

Image by Pexels from Pixabay

About me

Andrea Dottor

Microsoft MVP Developer Technologies

```
{
    site:        www.dottor.net
    email:       andrea@dottor.net
    twitter:     @dottor
    mastodon:    @dottor@hachyderm.io
}
```