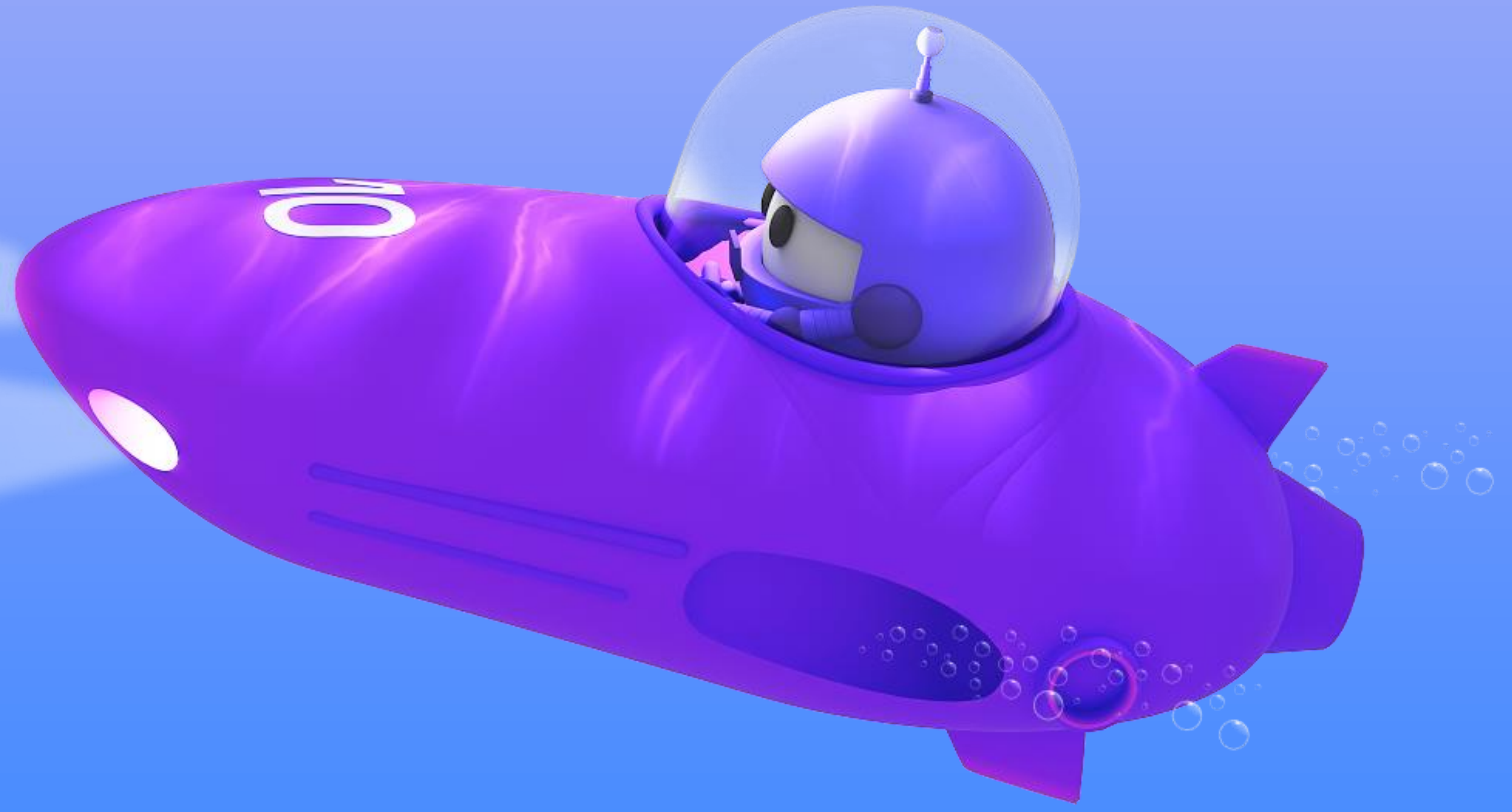
San Antonio .NET Conf 2025

.NET



SAN ANTONIO .NET

AUSTIN .NET

Daniel Ward

Dev, consultant

Microsoft MVP - .NET

Sponsors

LEAN TECHNIQUES

https://leantechniques.com

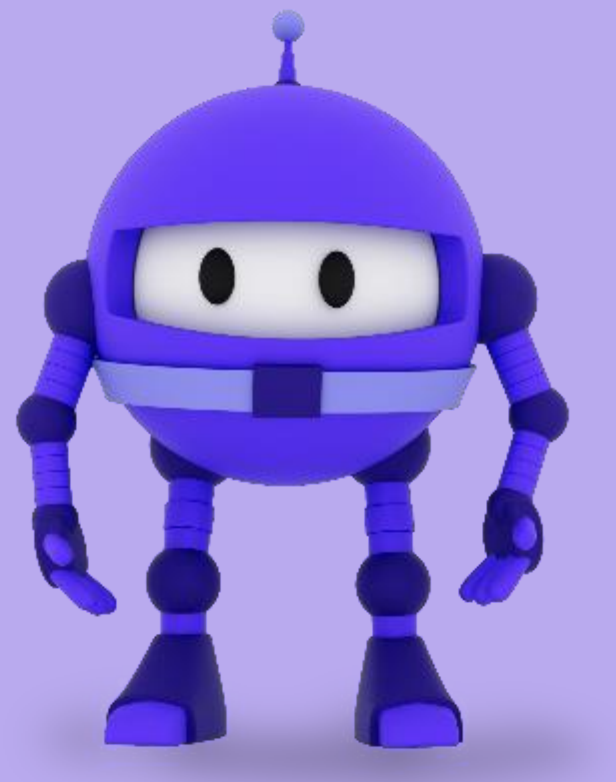
Sponsors

LEAN
TECHNIQUES

https://leantechniques.com

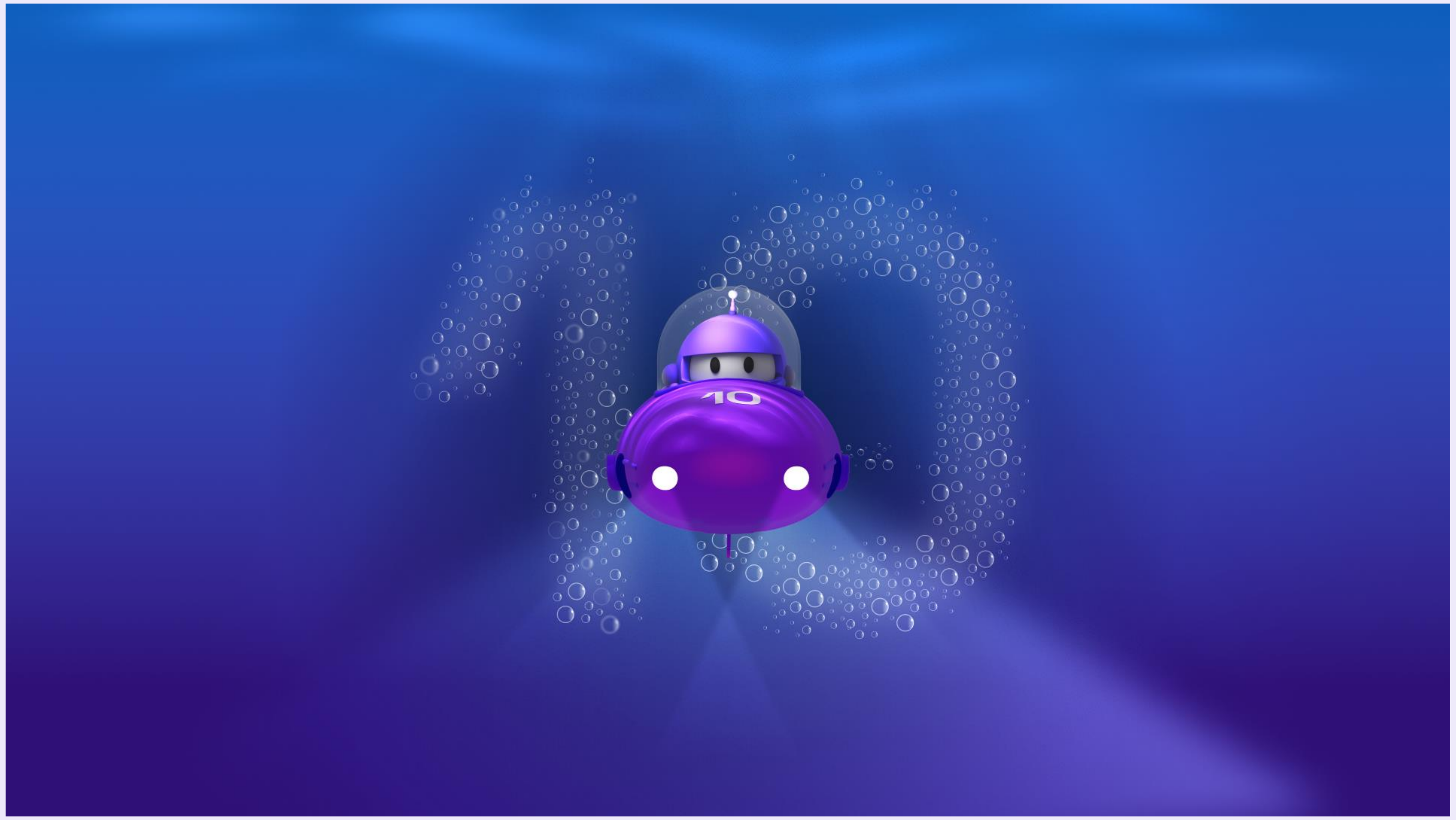
Outline

01 .NET 10



.NET 10

- Even number – LTS release
- Supported for 3 years
- Support ends Nov 14, 2028



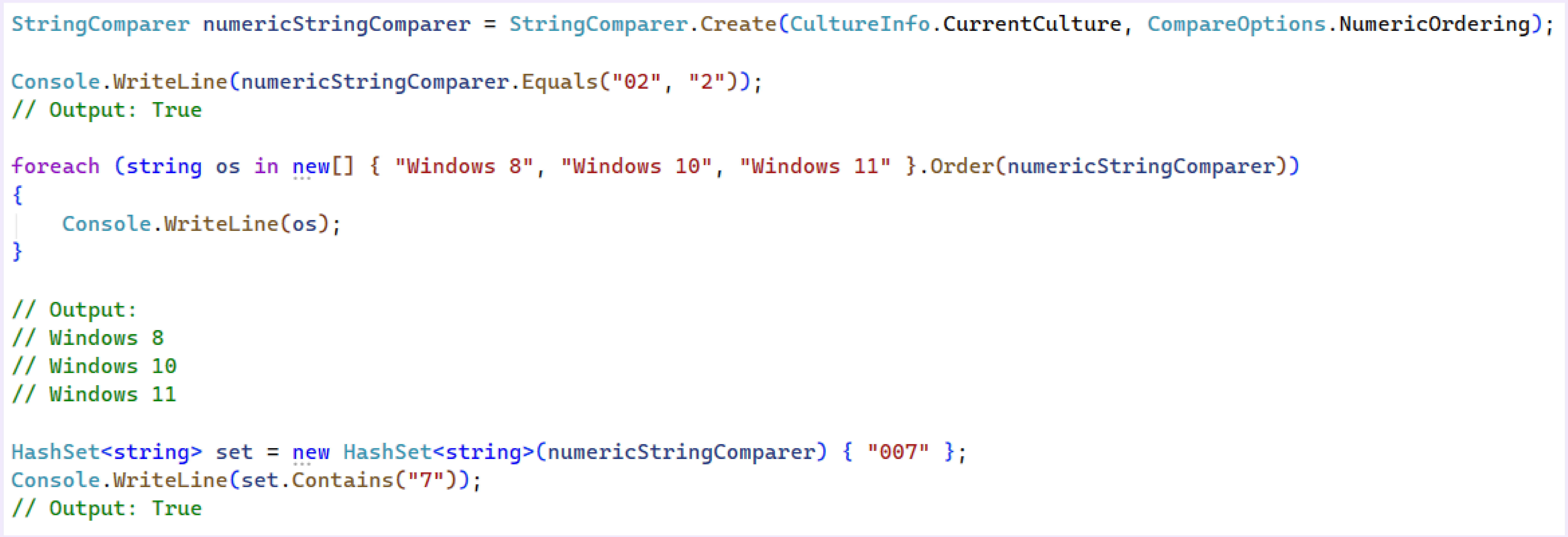
Performance

Performance improvements. Lots of them.

- https://devblogs.microsoft.com/dotnet/performance-improvements-in-net-10/

String comparison

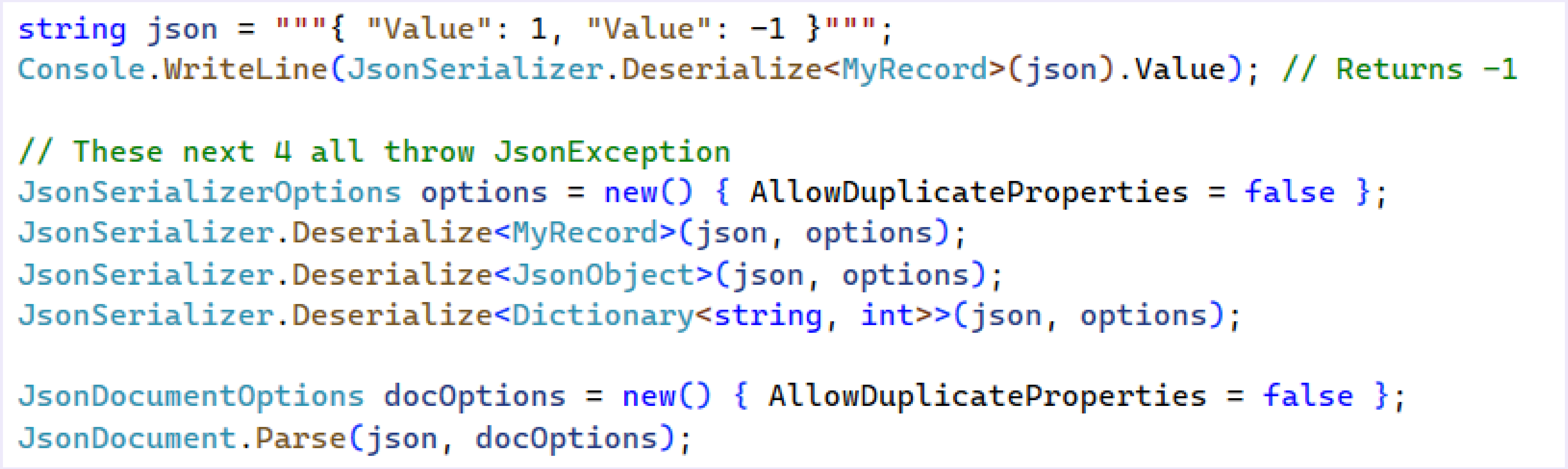
Numerical string comparison support



```csharp
StringComparer numericStringComparer = StringComparer.Create(CultureInfo.CurrentCulture, CompareOptions.NumericOrdering);

Console.WriteLine(numericStringComparer.Equals("02", "2"));
// Output: True

foreach (string os in new[] { "Windows 8", "Windows 10", "Windows 11" }.Order(numericStringComparer))
{
    Console.WriteLine(os);
}

// Output:
// Windows 8
// Windows 10
// Windows 11

HashSet<string> set = new HashSet<string>(numericStringComparer) { "007" };
Console.WriteLine(set.Contains("7"));
// Output: True
```

JSON options

Now supports disallowing duplicate JSON properties



```
string json = """{ "Value": 1, "Value": -1 }""";
Console.WriteLine(JsonSerializer.Deserialize<MyRecord>(json).Value); // Returns -1

// These next 4 all throw JsonException
JsonSerializerOptions options = new() { AllowDuplicateProperties = false };
JsonSerializer.Deserialize<MyRecord>(json, options);
JsonSerializer.Deserialize<JsonObject>(json, options);
JsonSerializer.Deserialize<Dictionary<string, int>>(json, options);

JsonDocumentOptions docOptions = new() { AllowDuplicateProperties = false };
JsonDocument.Parse(json, docOptions);
```

JSON options

New preset for stricter default options
JsonSerializerOptions.Strict that follows best practices

- No unmapped members allowed
- No duplicate properties allowed
- Preserves case-sensitive binding
- Respects nullable annotations
- Respects required constructor parameters

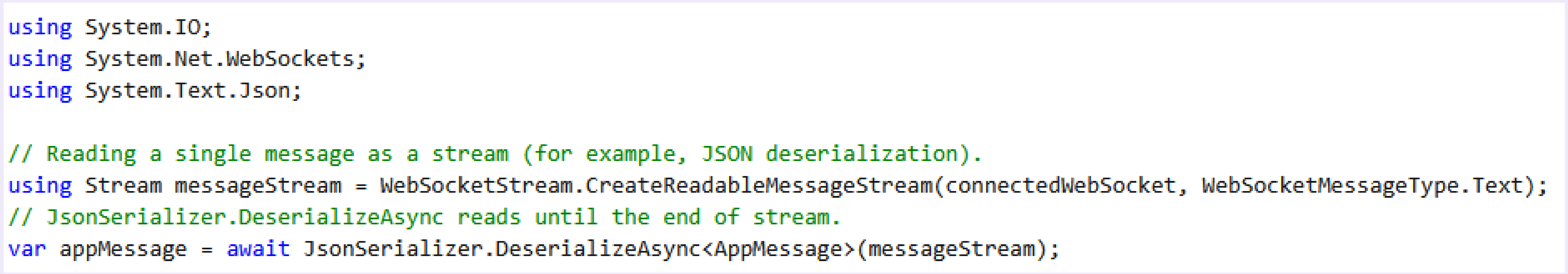
ZIP APIs

- Async ZIP APIs!
- Improved performance + memory usage of ZipArchive

WebSocketStream

WebSocketStream, a new API to simplify common WebSocket scenarios

- Abstracts details into a normal Stream object



```
using System.IO;
using System.Net.WebSockets;
using System.Text.Json;

// Reading a single message as a stream (for example, JSON deserialization).
using Stream messageStream = WebSocketStream.CreateReadableMessageStream(connectedWebSocket, WebSocketMessageType.Text);
// JsonSerializer.DeserializeAsync reads until the end of stream.
var appMessage = await JsonSerializer.DeserializeAsync<AppMessage>(messageStream);
```

dnx

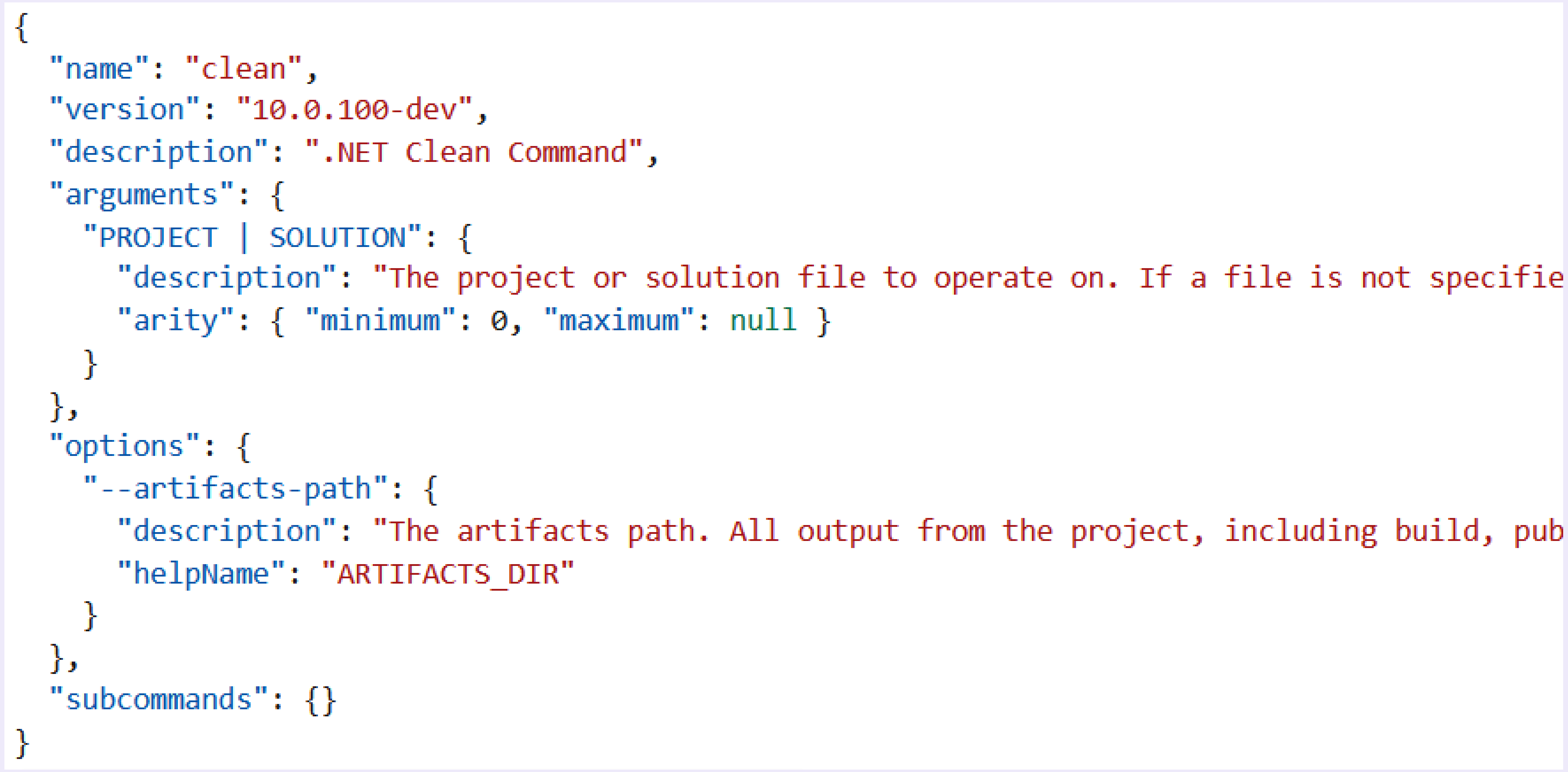
New command for one-shot running dotnet tools without installing them

- Dotnet tool exec – command that one-shot executes a tool
  - dotnet tool exec source ./artifacts/package dotnetsay “Hello, world!”
- dnx – shell script that calls dotnet tool exec, passing the arguments
  - dnx dotnetsay “Hello, world!”

CLI schema

New --cli-schema option on all CLI commands

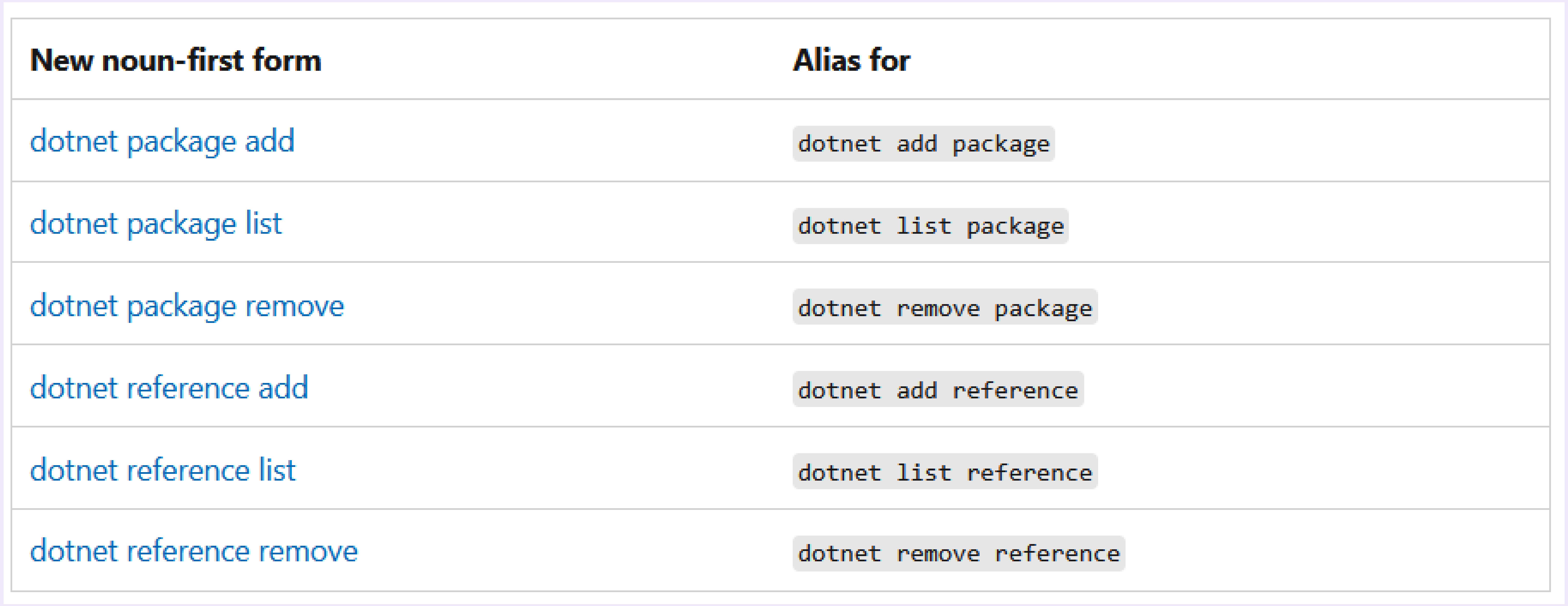
- dotnet clean --cli-schema



```
{
  "name": "clean",
  "version": "10.0.100-dev",
  "description": ".NET Clean Command",
  "arguments": {
    "PROJECT | SOLUTION": {
      "description": "The project or solution file to operate on. If a file is not specifie
      "arity": { "minimum": 0, "maximum": null }
    }
  },
  "options": {
    "--artifacts-path": {
      "description": "The artifacts path. All output from the project, including build, pub
      "helpName": "ARTIFACTS_DIR"
    }
  },
  "subcommands": {}
}
```

CLI aliases

New dotnet command aliases to make them easier to call/remember



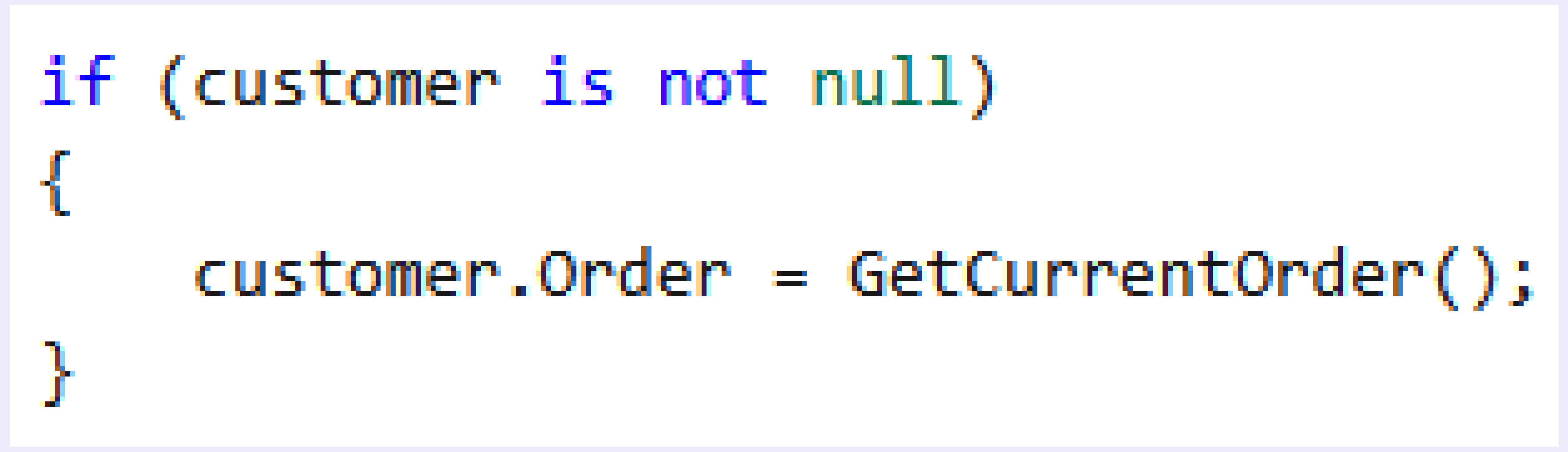
New noun-first form	Alias for
dotnet package add	`dotnet add package`
dotnet package list	`dotnet list package`
dotnet package remove	`dotnet remove package`
dotnet reference add	`dotnet add reference`
dotnet reference list	`dotnet list reference`
dotnet reference remove	`dotnet remove reference`

02 C# 14

Null-conditional assignments

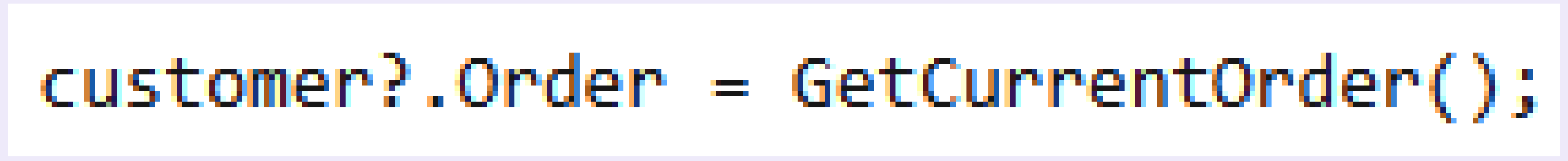
Shorthand to assign a member a value only if its object is not null

Before C# 14:



```
if (customer is not null)
{
    customer.Order = GetCurrentOrder();
}
```

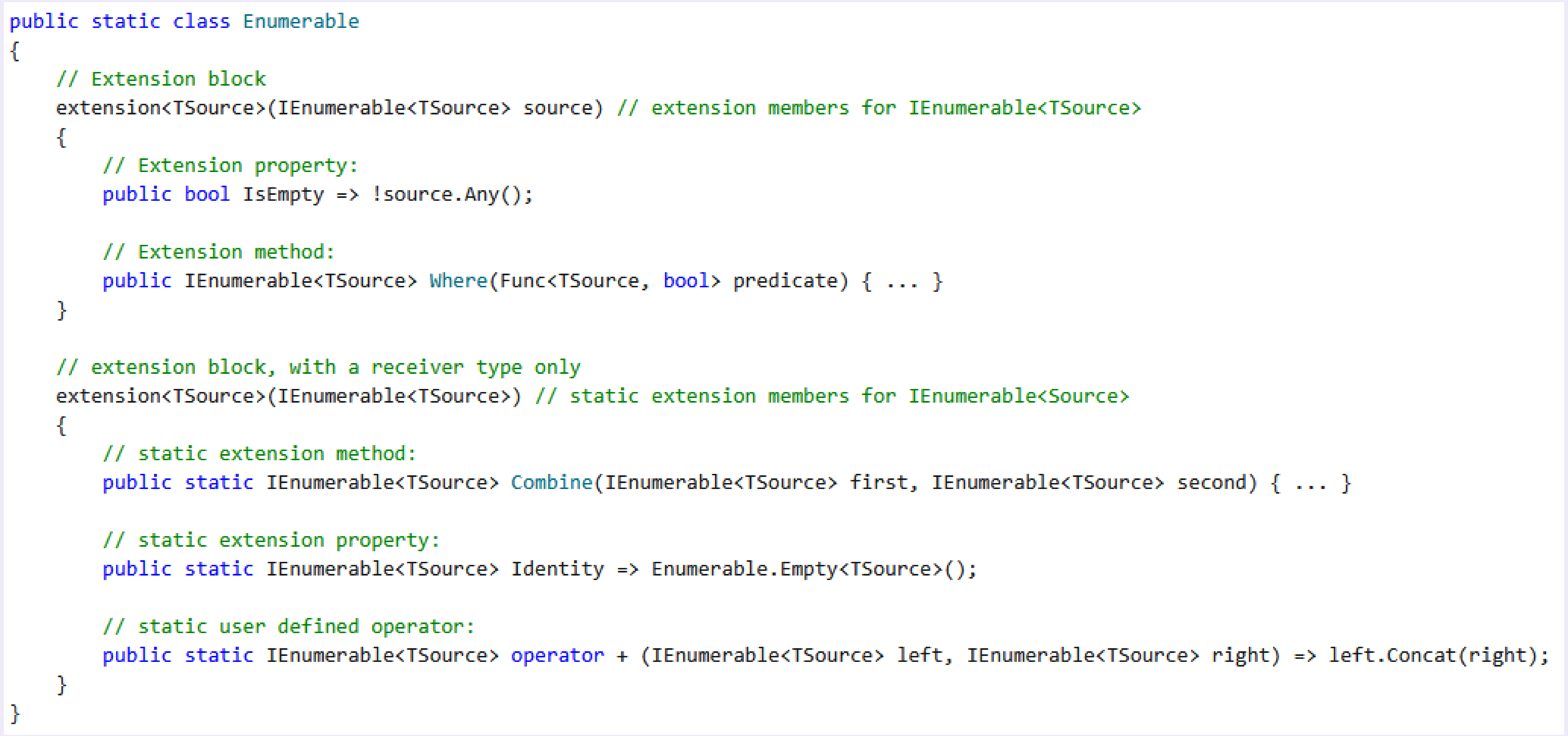
With C# 14:



```
customer?.Order = GetCurrentOrder();
```

Extension members

Now you can also extend properties + static methods!

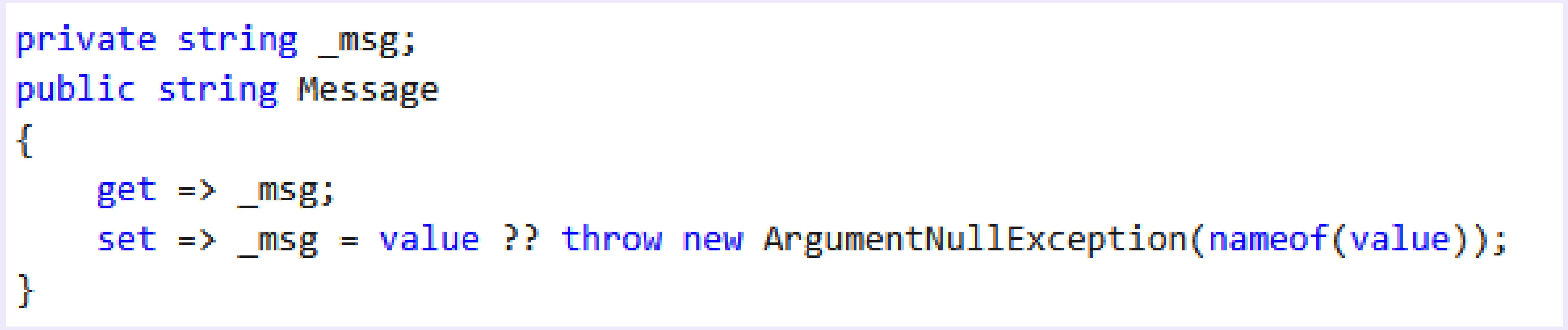


```csharp
public static class Enumerable
{
    // Extension block
    extension<TSource>(IEnumerable<TSource> source) // extension members for IEnumerable<TSource>
    {
        // Extension property:
        public bool IsEmpty => !source.Any();

        // Extension method:
        public IEnumerable<TSource> Where(Func<TSource, bool> predicate) { ... }
    }

    // extension block, with a receiver type only
    extension<TSource>(IEnumerable<TSource>) // static extension members for IEnumerable<Source>
    {
        // static extension method:
        public static IEnumerable<TSource> Combine(IEnumerable<TSource> first, IEnumerable<TSource> second) { ... }

        // static extension property:
        public static IEnumerable<TSource> Identity => Enumerable.Empty<TSource>();

        // static user defined operator:
        public static IEnumerable<TSource> operator + (IEnumerable<TSource> left, IEnumerable<TSource> right) => left.Concat(right);
    }
}
```

Field keyword

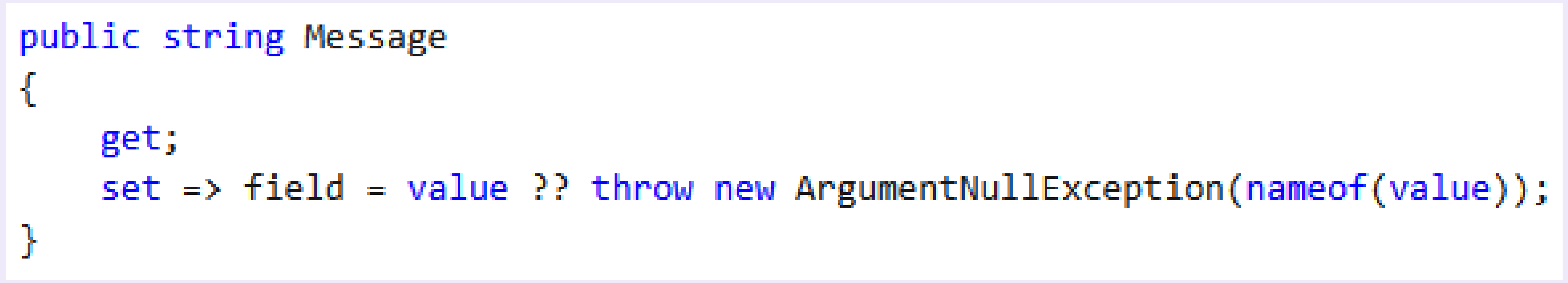
New field keyword allows you to write a property accessor body without a backing field

Before C# 14:



```
private string _msg;
public string Message
{
    get => _msg;
    set => _msg = value ?? throw new ArgumentNullException(nameof(value));
}
```

With C# 14:

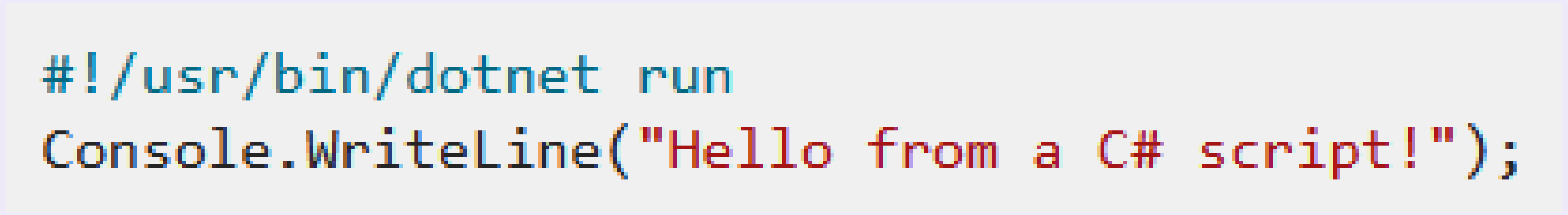


```
public string Message
{
    get;
    set => field = value ?? throw new ArgumentNullException(nameof(value));
}
```

File-based apps

Run a C# file directly with dotnet publish myapp.cs

- Targets native AOT by default
- Can use #:project, #:package, #:sdk for references



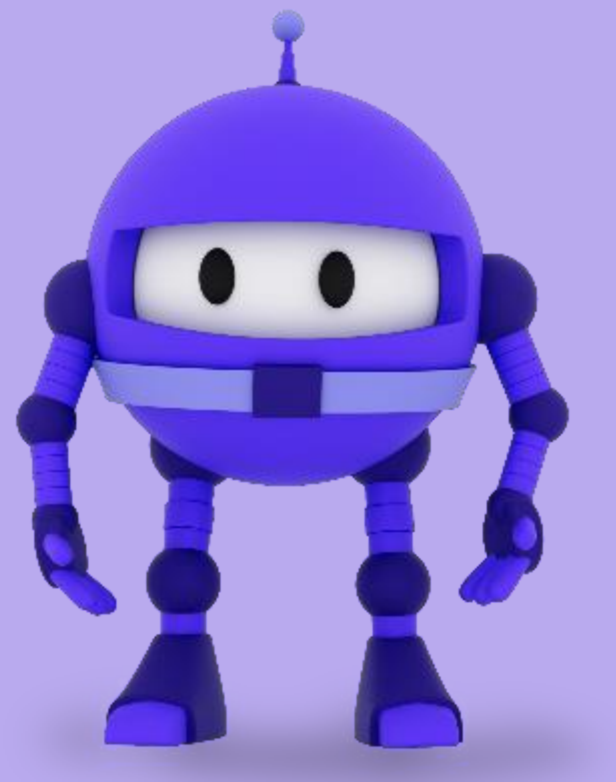
```
#!/usr/bin/dotnet run
Console.WriteLine("Hello from a C# script!");
```

File-based apps

```csharp
#:sdk Microsoft.NET.Sdk.Web
#:package Microsoft.AspNetCore.OpenApi@10.*-*

var builder = WebApplication.CreateBuilder();

builder.Services.AddOpenApi();

var app = builder.Build();

app.MapGet("/", () => "Hello, world!");
app.Run();
```

```xml
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net10.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="10.*-*" />
  </ItemGroup>

</Project>
```

03 EF Core



Azure SQL & SQL Server

Now fully supports vector & json data types

- More popular now with AI

Complex types - optionals

Complex types now support optional types

```
modelBuilder.Entity<Customer>(b =>
{
    b.ComplexProperty(c => c.ShippingAddress);
    b.ComplexProperty(c => c.BillingAddress);
});
```

```
CREATE TABLE [Customers] (
    [Id] int NOT NULL IDENTITY,
    [Name] nvarchar(max) NOT NULL,
    [BillingAddress_City] nvarchar(max) NOT NULL,
    [BillingAddress_PostalCode] nvarchar(max) NOT NULL,
    [BillingAddress_Street] nvarchar(max) NOT NULL,
    [BillingAddress_StreetNumber] int NOT NULL,
    [ShippingAddress_City] nvarchar(max) NOT NULL,
    [ShippingAddress_PostalCode] nvarchar(max) NOT NULL,
    [ShippingAddress_Street] nvarchar(max) NOT NULL,
    [ShippingAddress_StreetNumber] int NOT NULL,
    CONSTRAINT [PK_Customers] PRIMARY KEY ([Id])
);
```

```
public class Customer
{
    ...

    public Address ShippingAddress { get; set; }
    public Address? BillingAddress { get; set; }
}
```

Complex types – mapping to JSON

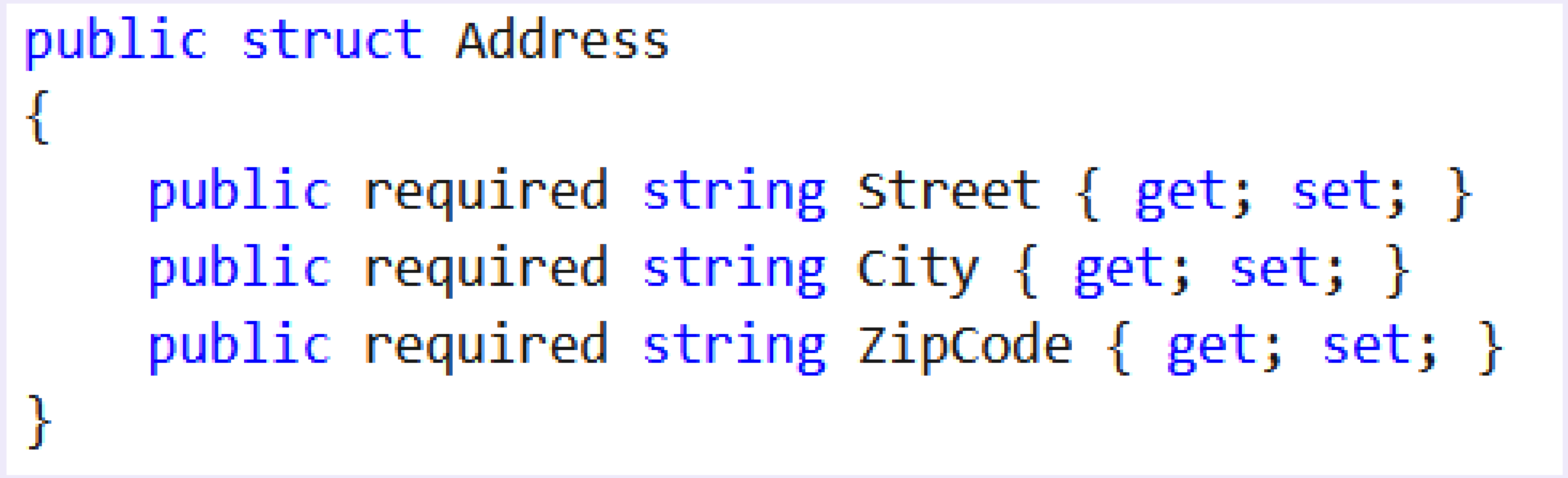
Complex types now support mapping to JSON

```
modelBuilder.Entity<Customer>(b =>
{
    b.ComplexProperty(c => c.ShippingAddress, c => c.ToJson());
    b.ComplexProperty(c => c.BillingAddress, c => c.ToJson());
});
```

```
CREATE TABLE [Customers] (
    [Id] int NOT NULL IDENTITY,
    [Name] nvarchar(max) NOT NULL,
    [ShippingAddress] json NOT NULL,
    [BillingAddress] json NOT NULL NULL,
    CONSTRAINT [PK_Customers] PRIMARY KEY ([Id])
);
```

Complex types – structs

Complex types now support structs

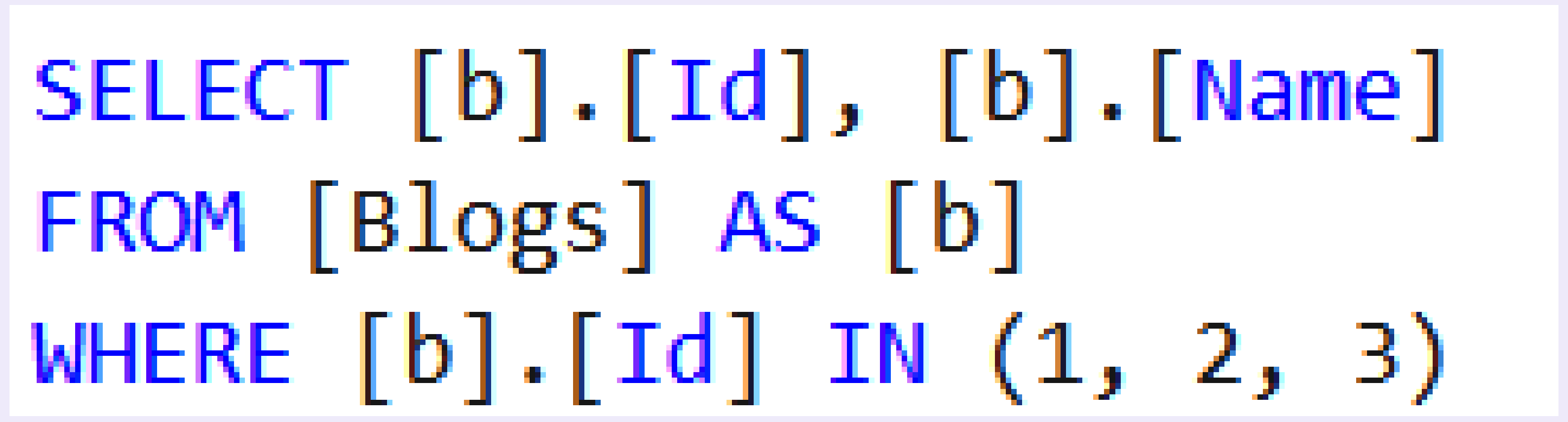


```
public struct Address
{
    public required string Street { get; set; }
    public required string City { get; set; }
    public required string ZipCode { get; set; }
}
```

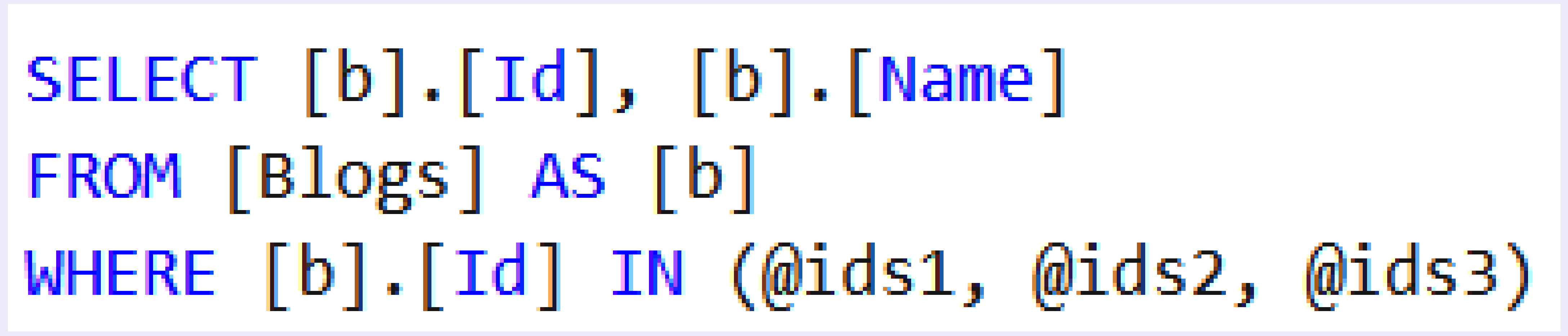
LINQ-SQL translation

Improved translation for parameterized collections

- Most highly-voted issue in the efcore repo at the time
- New default translation mode for parameterized collections. Each value in the collection becomes its own scalar parameter
- TL;DR: save the CPU on your DB server



```sql
SELECT [b].[Id], [b].[Name]
FROM [Blogs] AS [b]
WHERE [b].[Id] IN (1, 2, 3)
```

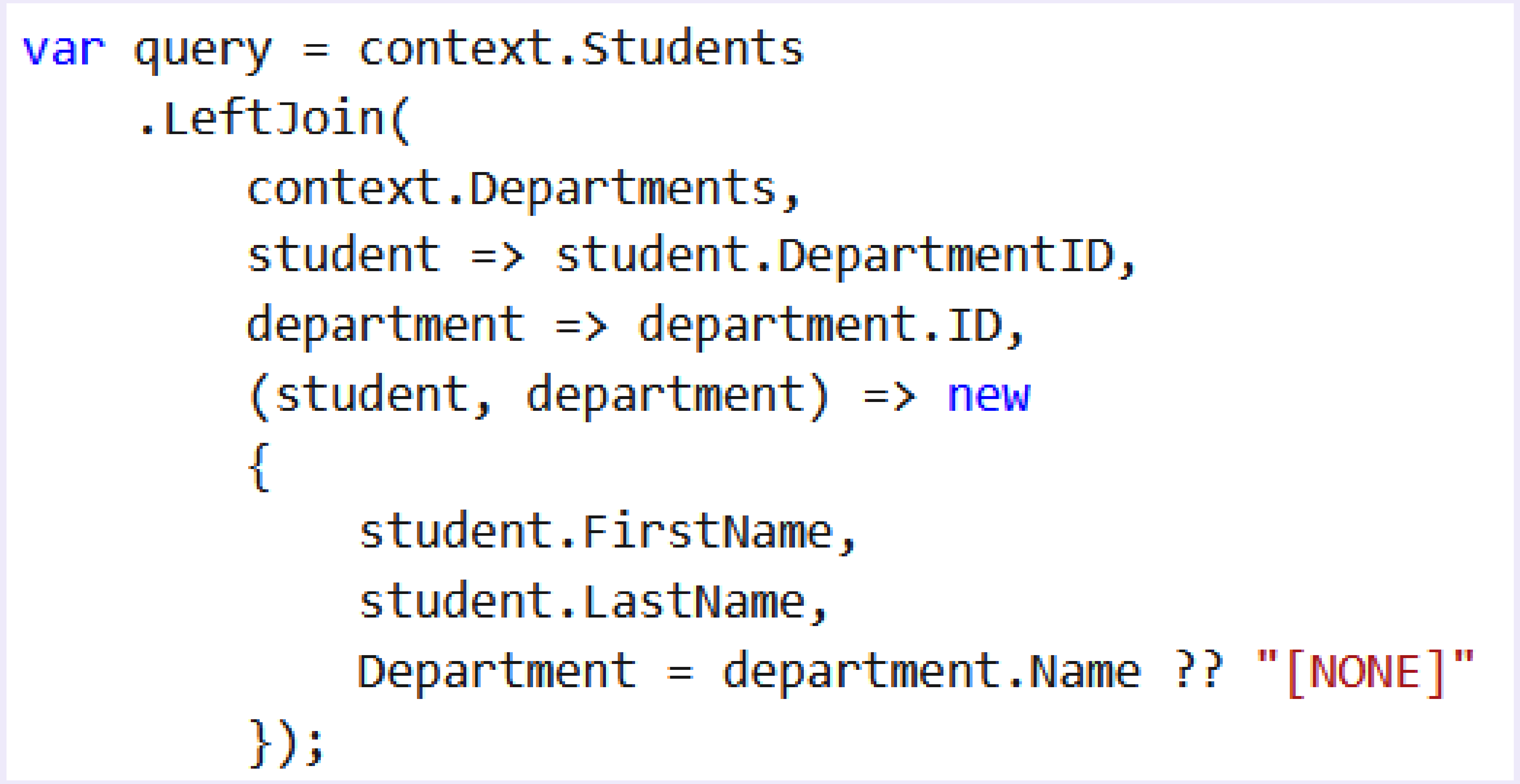


```sql
SELECT [b].[Id], [b].[Name]
FROM [Blogs] AS [b]
WHERE [b].[Id] IN (@ids1, @ids2, @ids3)
```

LeftJoin & RightJoin

First-class support for left and right join!

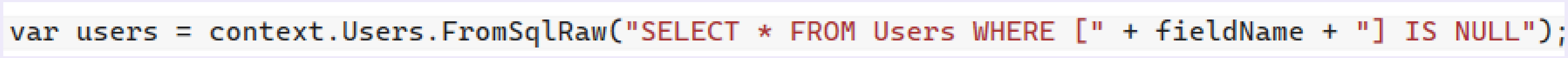
- Was possible before, but you had to use SelectMany, GroupJoin, and DefaultIfEmpty



```
var query = context.Students
    .LeftJoin(
        context.Departments,
        student => student.DepartmentID,
        department => department.ID,
        (student, department) => new
        {
            student.FirstName,
            student.LastName,
            Department = department.Name ?? "[NONE]"
        });
```

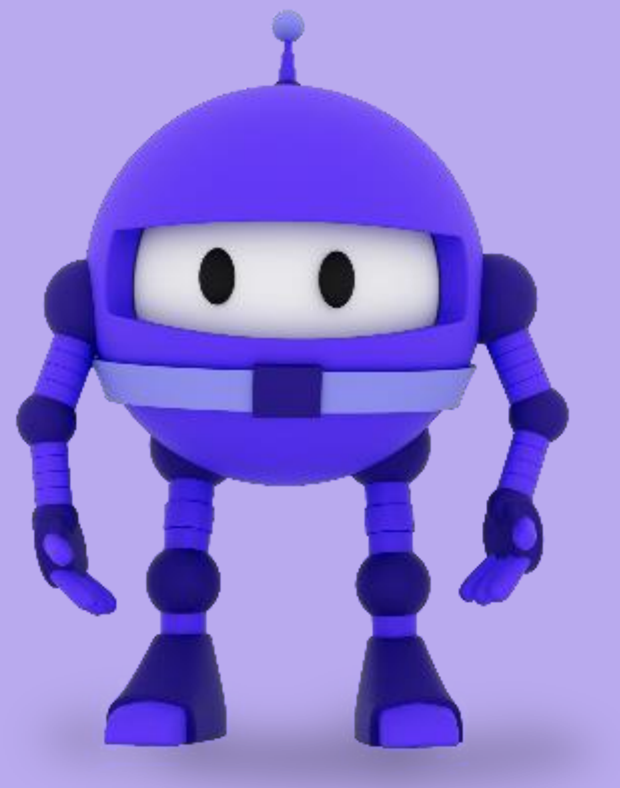
Security

- No longer logs inlined constants from queries
- FromSqlRaw now has an analyzer that warns if concatenation is performed within a raw SQL method
  - Can be suppressed if you trust the field and/or have sanitized it



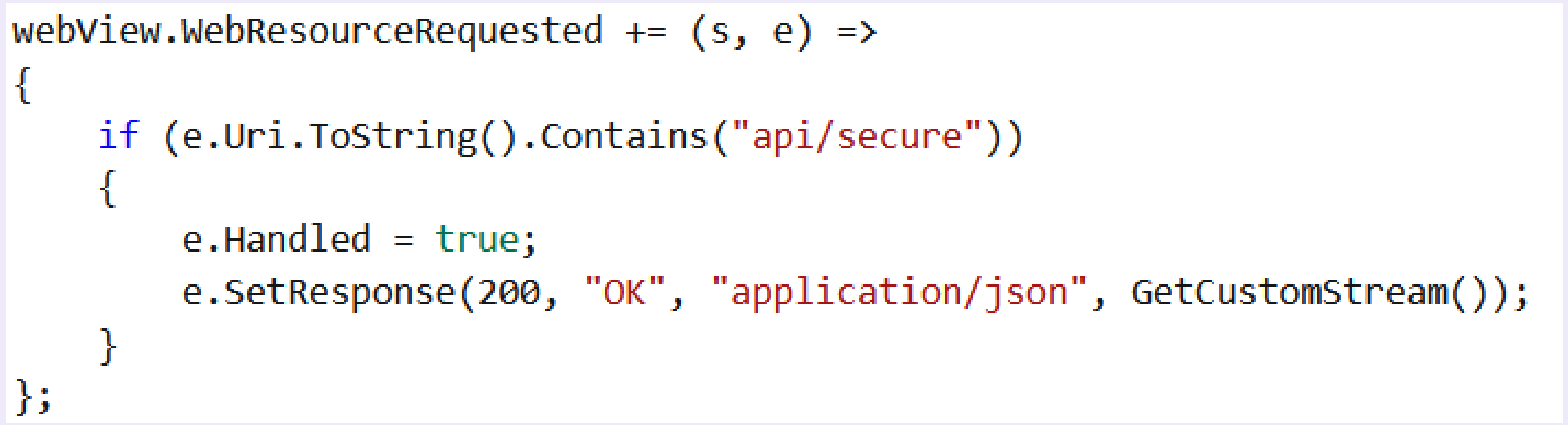
```
var users = context.Users.FromSqlRaw("SELECT * FROM Users WHERE [" + fieldName + "] IS NULL");
```

04 MAUI + WPF + WinForms



MAUI

- Lots and lots and lots of small changes to various controls, services, etc.
- Integration with Aspire
  - builder.AddServiceDefaults()
- Now now intercept and respond to web requests from BlazorWebView and HybridWebView



```
webView.WebResourceRequested += (s, e) =>
{
    if (e.Uri.ToString().Contains("api/secure"))
    {
        e.Handled = true;
        e.SetResponse(200, "OK", "application/json", GetCustomStream());
    }
};
```

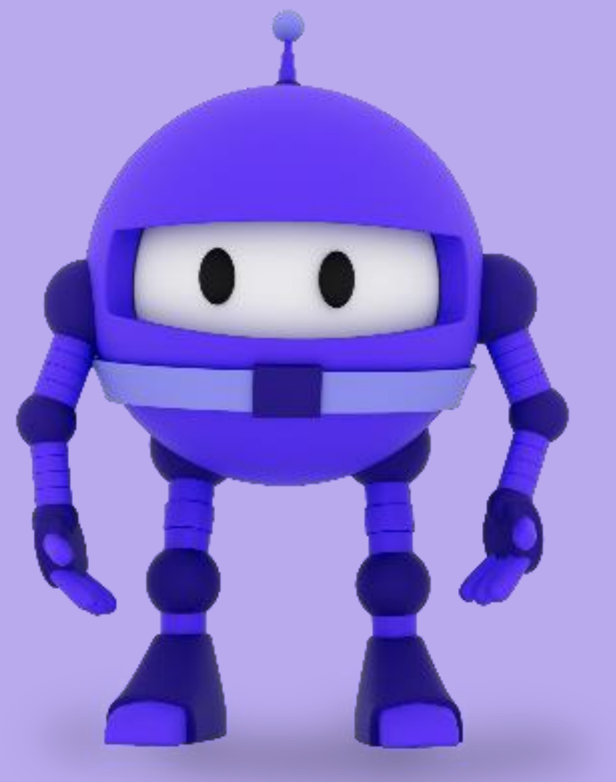
WPF

- General performance improvements
- Bug fixes & code cleanup
- Fluent styles added to more controls
- WPF and WinForms now use the same clipboard API

WinForms

- Bug fixes & code cleanup
- Full async forms support, no longer experimental
- Full dark mode support, no longer experimental
- New API Form.ScreenCaptureMode to prevent screen cap apps from capturing a form (if that app uses the Windows API)
  - Useful for protecting sensitive info (ids, passwords, etc.)

05 AI + Visual Studio



AI

Microsoft Agent Framework

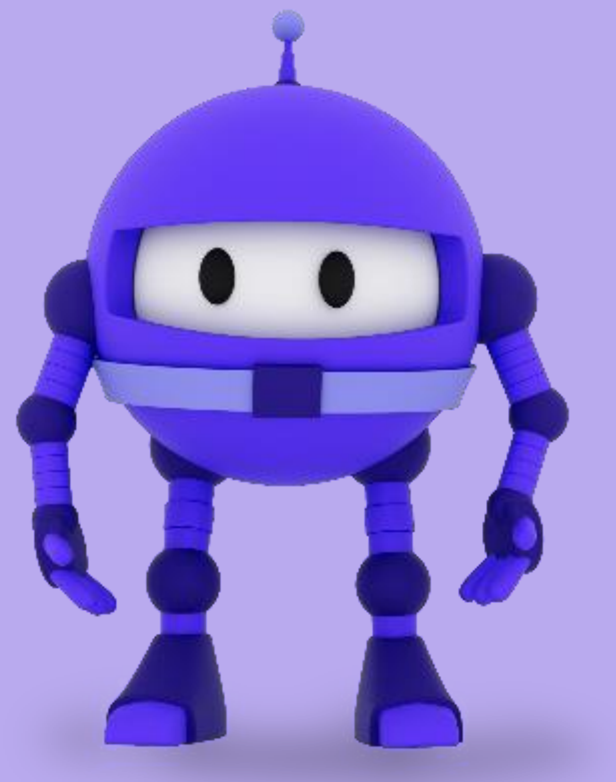
- Consolidates Semantic Kernel and AutoGen into a single library

Visual Studio 2026 Insiders

New Visual Studio that came out in October

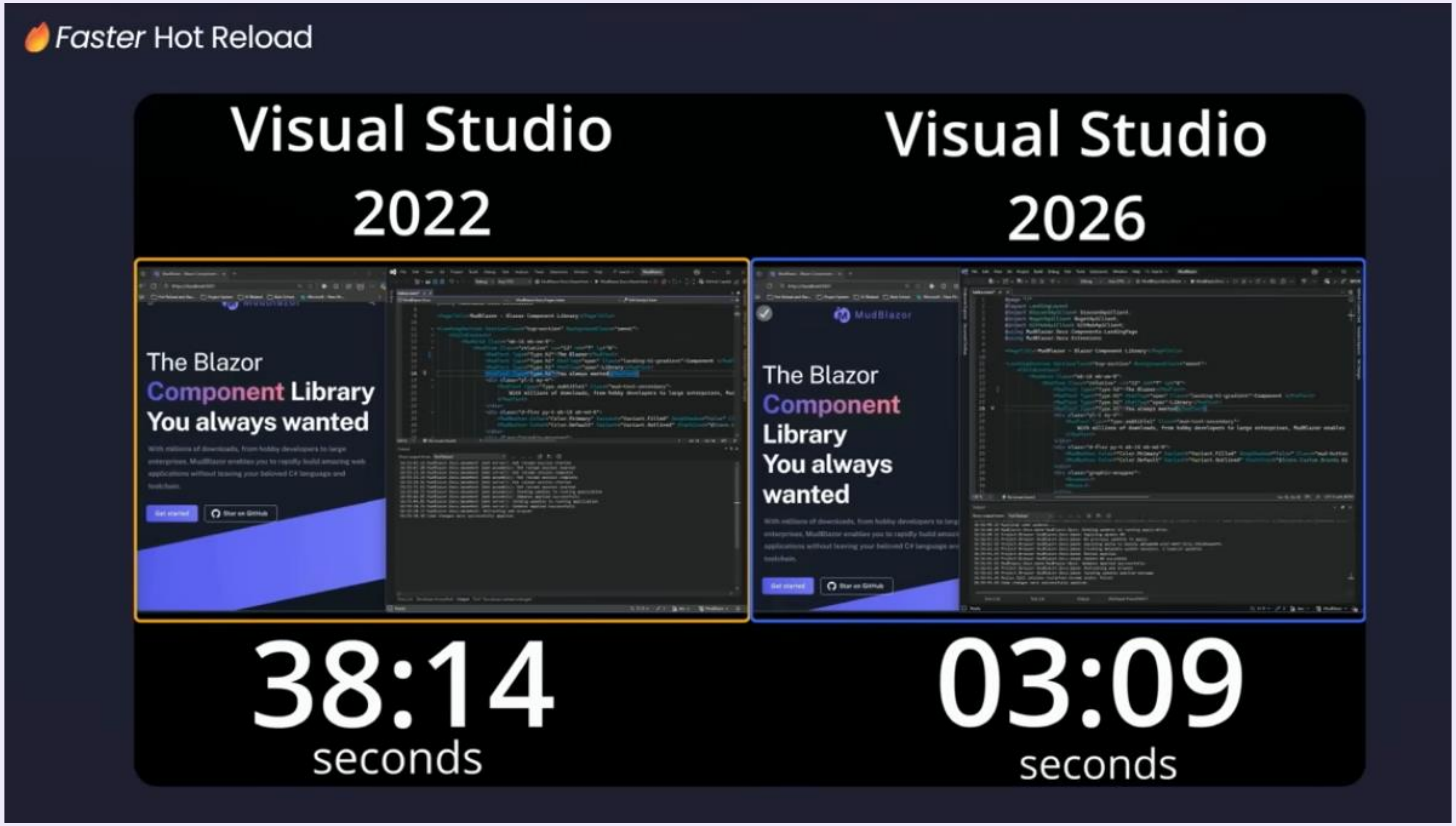
- Adaptive paste
- Profiler Copilot Agent
- Debugger Agent
- Mermaid chart rendering
- Code coverage now available in all editions
- BenchmarkDotNet project template

06 ASP.NET Core



Blazor – hot reload

Faster hot reload in Visual Studio, even with large applications using libraries like MudBlazor



Blazor – script as static asset

Blazor script now served automatically as a static asset

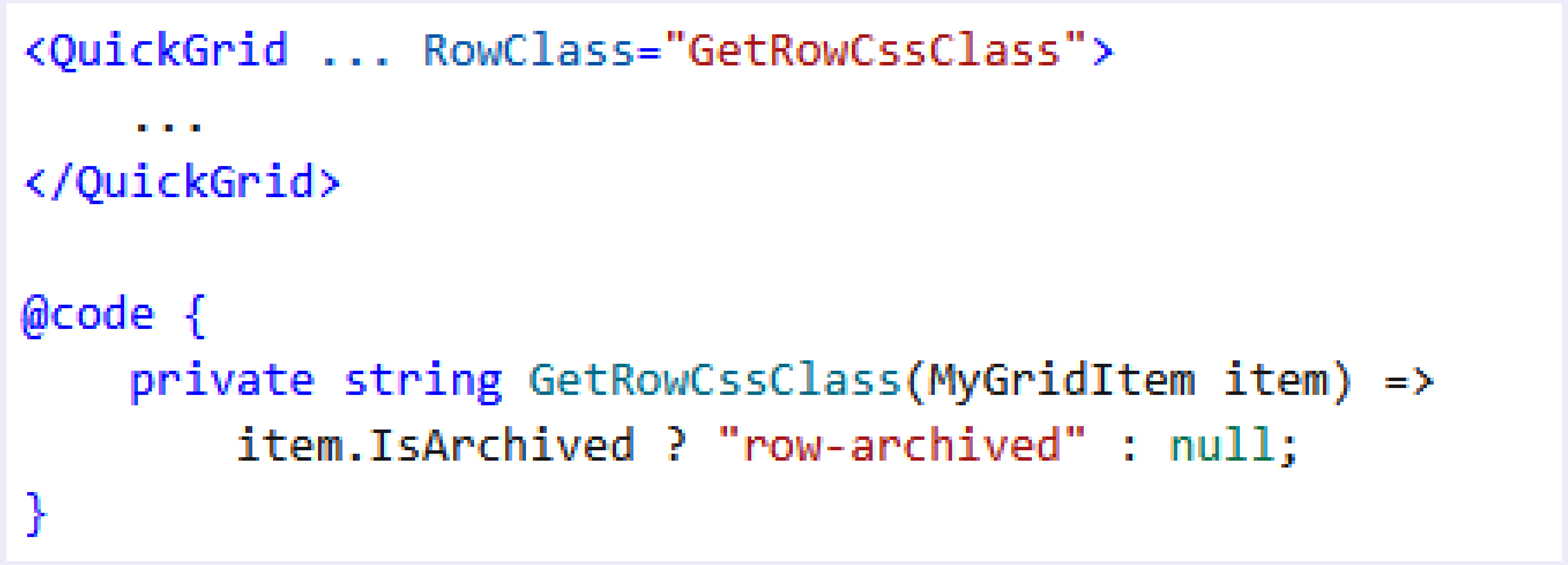
- Allows for better caching, cache busting, compression

Blazor – client-side fingerprinting

- .NET 9 introduced server-side fingerprinting for Blazor Web Apps
- You can now opt into client-side fingerprinting for standalone Blazor WebAssemblyApps
- Placeholders in index.html are overridden with values computed during build

Blazor – general 1/2

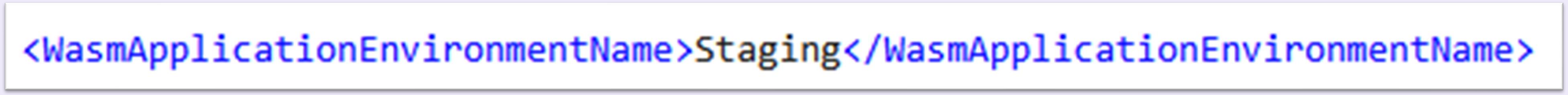
- New how-to samples for OIDC, Entra ID, and Windows Authentication
  - Including BFF pattern for OIDC and Entra ID
- New RowClass attribute on QuickGrid to apply a stylesheet class to a row



```
<QuickGrid ... RowClass="GetRowCssClass">
    ...
</QuickGrid>

@code {
    private string GetRowCssClass(MyGridItem item) =>
        item.IsArchived ? "row-archived" : null;
}
```

Blazor – general 2/2

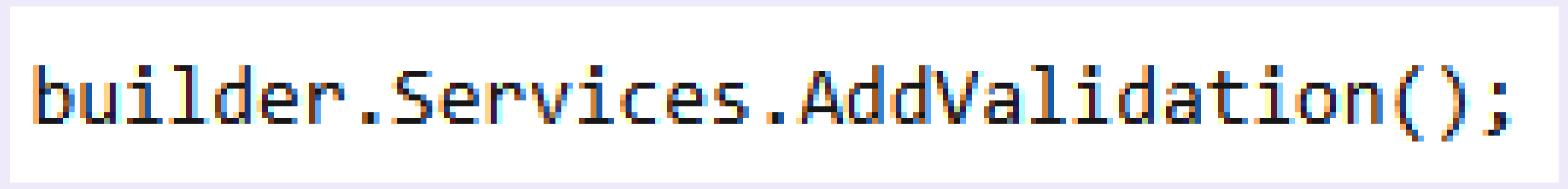
- Comprehensive metrics + tracing capabilities (eg. component lifecycle, event handling)
- WebAuthn (passkey) support with ASP.NET Core Identity
- Standalone WASM apps (aka no .NET server backend) no longer use launchsettings.json for the environment



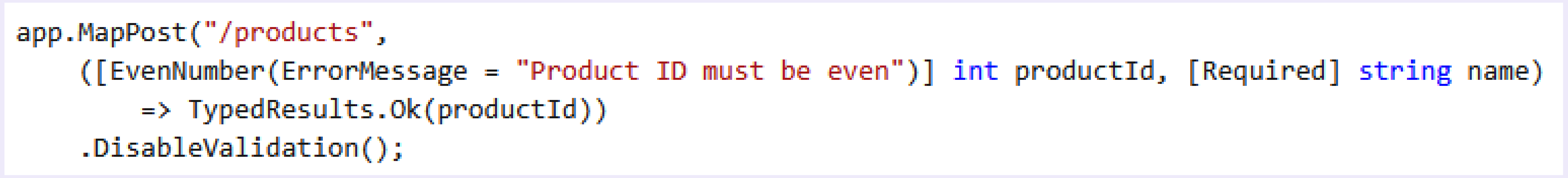
```xml
<WasmApplicationEnvironmentName>Staging</WasmApplicationEnvironmentName>
```

Minimal APIs

- Validation support – builder.Services.AddValidation()
  - – Can use data annotations, FluentValidation, etc.



```
builder.Services.AddValidation();
```



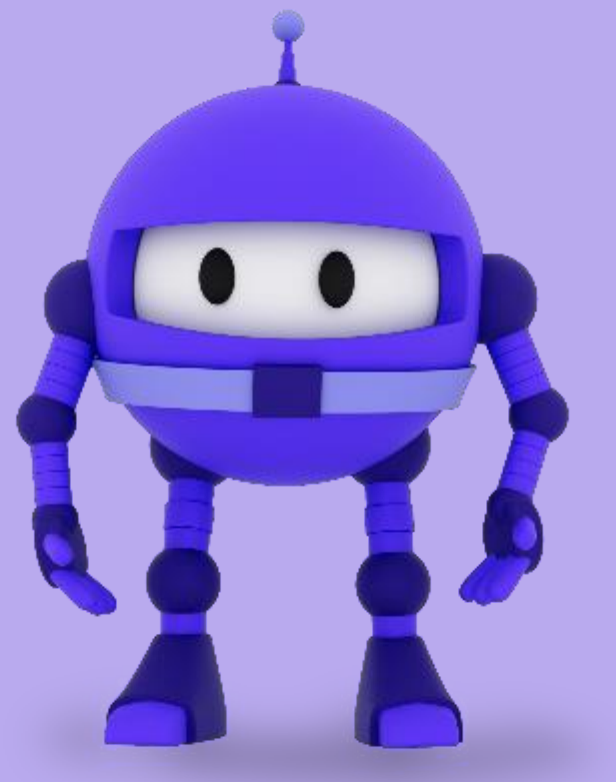
```
app.MapPost("/products",
    ([EvenNumber(ErrorMessage = "Product ID must be even")] int productId, [Required] string name)
        => TypedResults.Ok(productId))
    .DisableValidation();
```

- Validation with records
- Server-sent events now supported

Miscellaneous ASP.NET Core stuff

- OpenAPI 3.1 support – big change to OpenAPI!
- Metrics flowing from authentication, authorization, Identity
- Better support for WebApplicationFactory – no longer have to make Program.cs a partial class
  - Program class is now public by default vs internal
- Support for .localhost top-level domain
- Automatic eviction of memory used by the web server itself (Kestrel, HTTP.sys, IIS)

07 Aspire



Aspire – general 1/2

- .NET Aspire is now Aspire
- Jumped from v9 to v13 to decouple its version number from the .NET version number to avoid confusion
- Support for C#, Python, and Javascript/Typescript apps
- Combine your .NET app with a React/Angular/Vue frontend of Python AI sidecar app
- Aspire replaces Docker Compose for local dev while also giving you a local dashboard with OTEL, structured logging, etc wired up for you

Aspire – general 2/2

- New CLI
- Cleaner starter templates
- MAUI integration
- MCP server included in the dashboard
- Application map improvements
- Aspire dashboard included by default when deploying to Azure App Service

Resources

- What’s new in .NET 10 on MS Learn
  - https://learn.microsoft.com/en-us/dotnet/core/whats-new/dotnet-10/overview
- Stephen Toub post on performance improvements
  - https://devblogs.microsoft.com/dotnet/performance-improvements-in-net-10/
- What’s new in Aspire 13
  - https://aspire.dev/whats-new/aspire-13/

Thank you!