



✨ What's New in .NET 8? ✨



Martin Costello

Principal Engineer @ Just Eat Takeaway.com
He/him/his

@martin_costello



@martincostello



What's on the menu?

- What is .NET 8? 🧑
- Themes of .NET 8 🎵
- What's new in .NET 8? NEW
- C# 12 📄
 - Collection expressions - []
 - Primary constructors - 🏆
- New APIs ✨
 - TimeProvider ⌚
 - Resilience libraries 🦜
- A brief interlude on AoT 🏃
- Source generators ☐
 - Request Delegate Generator ⚡
 - Configuration Binding 🦸
- SDK features 🧑
 - Artifacts output 📦
 - Terminal logger 🪵





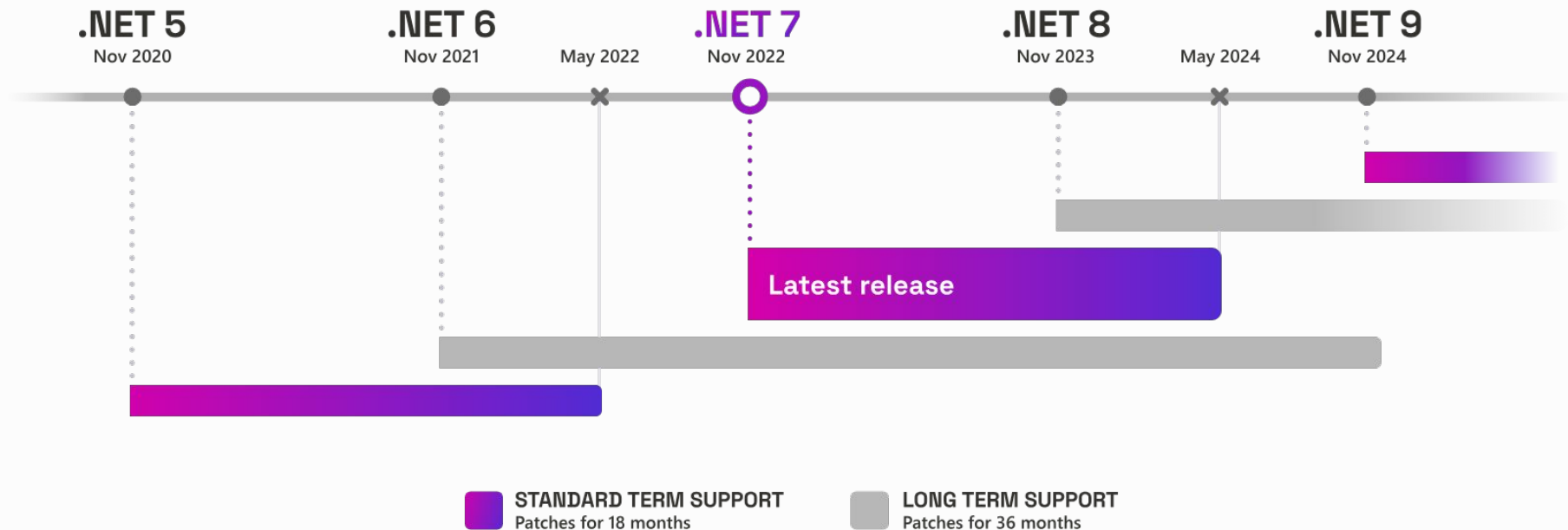
What is .NET 8? 🙋

What is .NET 8? 🙋



- The next major release of .NET (née .NET Core)
- Released on 14th November - hot off the presses 🔥
- The latest *Long Term Support* (LTS) release
- End-of-life countdown started for previous releases:
 - .NET 7 end-of-life: **14th May 2024**
 - .NET 6 end-of-life: **12th November 2024**

What is .NET 8? 🙋



[.NET and .NET Core Support Policy](#)

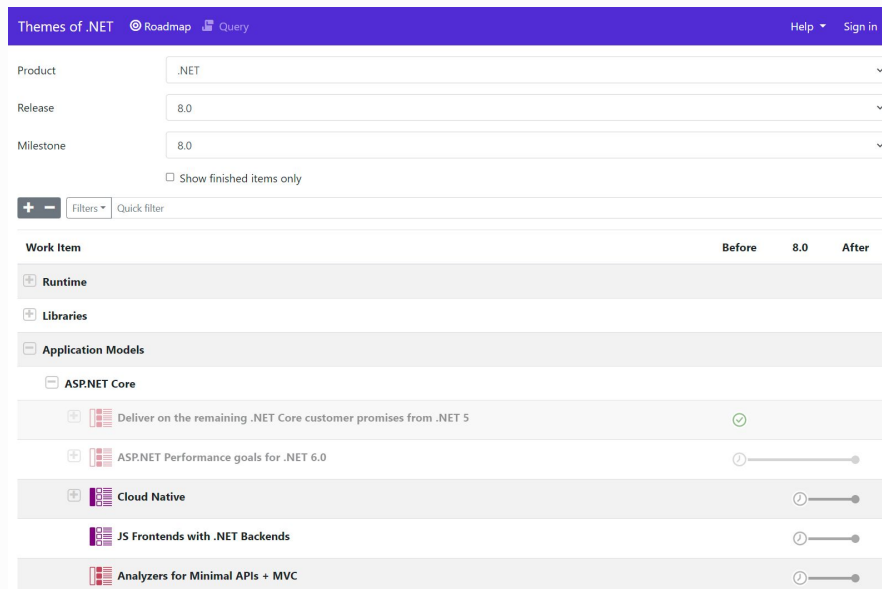


Themes of .NET 8

Themes of .NET 8



- Runtime
 - ARM64 - Prepare for the future
- Application Models
 - ASP.NET Core
 - Cloud Native
 - JS Frontends with .NET Backends
 - MAUI
 - Delight developers building client and mobile applications with .NET 8
- Tooling
 - Secure Supply Chain with NuGet



The screenshot shows the 'Themes of .NET' website with a purple header. The main content area has filters for Product (.NET), Release (8.0), and Milestone (8.0). Below the filters, there's a 'Show finished items only' checkbox and a 'Filters' button. The main table lists work items with columns for 'Work Item', 'Before', '8.0', and 'After'. The items are categorized under 'Runtime', 'Libraries', and 'Application Models'. The 'Application Models' category is expanded, showing 'ASP.NET Core' with two items: 'Deliver on the remaining .NET Core customer promises from .NET 5' (marked with a green check) and 'ASP.NET Performance goals for .NET 6.0' (marked with a progress bar). Below this, 'Cloud Native' is shown with a progress bar, followed by 'JS Frontends with .NET Backends' and 'Analyzers for Minimal APIs + MVC', both with progress bars.

Work Item	Before	8.0	After
Runtime			
Libraries			
Application Models			
ASP.NET Core			
Deliver on the remaining .NET Core customer promises from .NET 5			✓
ASP.NET Performance goals for .NET 6.0			1
Cloud Native			1
JS Frontends with .NET Backends			1
Analyzers for Minimal APIs + MVC			1

themesof.net



What's new in .NET 8?

NEW

What's new in .NET 8?



- C# 12
- Runtime
 - Improvements to System.Text.Json
 - More supported types e.g. Half, (U)Int128
 - Source generator improvements
 - snake_case and kebab-case naming
 - **Time abstraction with TimeProvider**
 - Frozen collections
 - Vector512<T> and AVX-512 support
 - SHA-3 support
- Extensions
 - Keyed DI services
 - Options validation source generator
 - IMeterFactory
- Native AoT
 - Support for macOS
 - Choose between performance and size

What's new in .NET 8?



- SDK
 - Evaluate MSBuild properties/items/target results
 - **Terminal build output**
 - **Simplified output paths**
 - dotnet workload clean
 - dotnet publish and pack for Release by default
 - dotnet restore security auditing
- Containers
 - Debian 12 (Bookworm 📖🔗)
 - Non-root user
 - New default ports - 8080/8081
- ASP.NET Core
 - Blazor full-stack web UI
 - Explicit [FromForm] binding for Minimal APIs
 - Antiforgery support for Minimal APIs
 - **AoT support for a subset of application types**
 - Built-in metrics for hosting, Kestrel and SignalR
- [.NET Aspire](#)

What's new in .NET 8?

NEW



- [What's new in .NET 8](#)
- [What's new in ASP.NET Core 8](#)
- [What's new in C# 12](#)





C# 12



New features:

- Primary constructors
- Collection expressions
- `ref readonly` parameters
- Inline arrays
- Optional lambda expression parameters
- Alias any type
- Experimental attribute
- *Interceptors* (preview)

[What's new in C# 12](#)



New features:

- **Primary constructors**
- **Collection expressions**
- `ref readonly` parameters
- Inline arrays
- Optional lambda expression parameters
- Alias any type
- Experimental attribute
- *Interceptors* (preview)

[What's new in C# 12](#)



Visual Studio demo   - [Sample code](#)



New APIs ✨

New APIs ✨



TimeProvider 🕒

- An abstraction for *the flow of time*
- Use cases
 - What is the current local/UTC date and time?
 - What is the current timestamp?
 - Create a timer that fires an event on a schedule
- Includes a fake implementation for tests
- [Documentation](#)



Resilience libraries 🦜

- New extensions libraries built on top of [Polly](#) v8
- Adding resilience with default settings for HttpClient can be as simple as adding **one line of code**.
- [Documentation](#)
- [Ensuring Resilience with Polly](#) 📺



New APIs ✨



Visual Studio demo 🧐💻 - [Sample code](#)



A brief interlude on AoT



A brief interlude on AoT




- AoT - Ahead of Time compilation
 - Opposite of JIT - Just in Time compilation
- Improves startup performance at the expense of increased compile times
- No runtime code generation via JIT
- No dynamic loading of assemblies (e.g. plugins)
- Types that are not used are trimmed away
- Many reflection use cases are not supported
 - An alternative to things like serializers is to use *source generators*...



Source generators ☐

Source generators



- [Source generators](#) were introduced in .NET 5 with C# 9
- Allow for dynamic code generation during compilation
- Existing examples include:
 - [JSON source generator](#)
 - [Regex source generator](#)
- Benefits
 - Move overhead from runtime to compile time
 - Work around runtime environment constraints...
 - ...such as AoT 

Source generators





Visual Studio demo   - [Sample code](#)

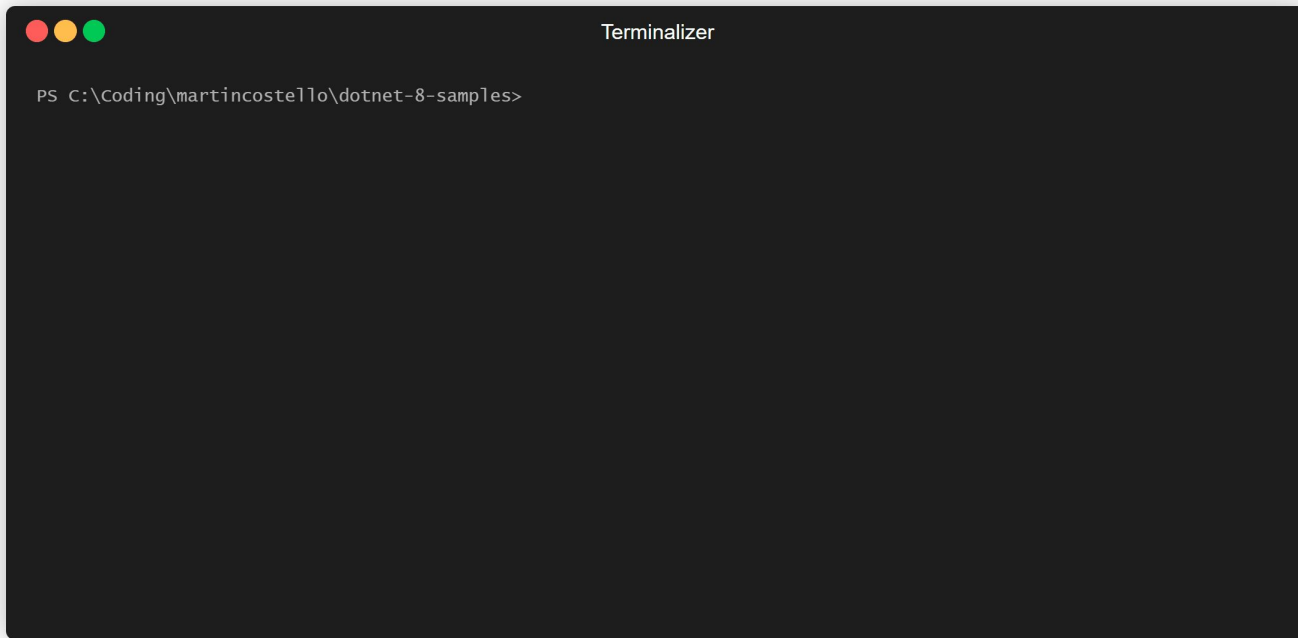


SDK features



- Artifacts output  - predictable location of build/publish/test output
- Terminal logger  - an easier visualisation of MSBuild outputs

SDK features 🧑 - Artifacts output 📦



Previous behaviour

SDK features 🧑 - Artifacts output 📦

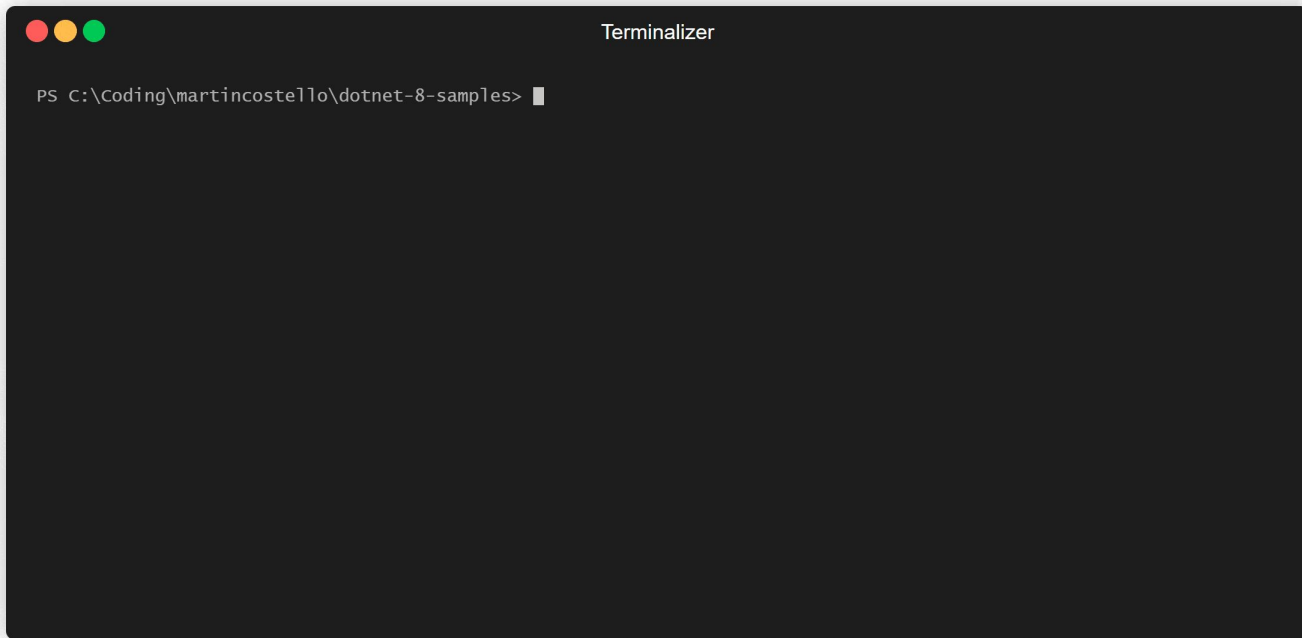


Output structure

- src
 - TimeProvider
 - bin
 - Release
 - net8.0
 - publish
 - CSharp12
 - bin
 - Release
 - net8.0
 - publish

Previous behaviour

SDK features 🧑 - Artifacts output 📦



Artifacts output - UseArtifactsOutput=true

SDK features - Artifacts output



Output structure


- artifacts
 - bin
 - CSharp12
 - release
 - TimeProvider
 - release
 - WeatherApi
 - release
 - publish
 - CSharp12
 - release
 - TimeProvider
 - release
 - WeatherApi
 - release

Artifacts output - UseArtifactsOutput=true

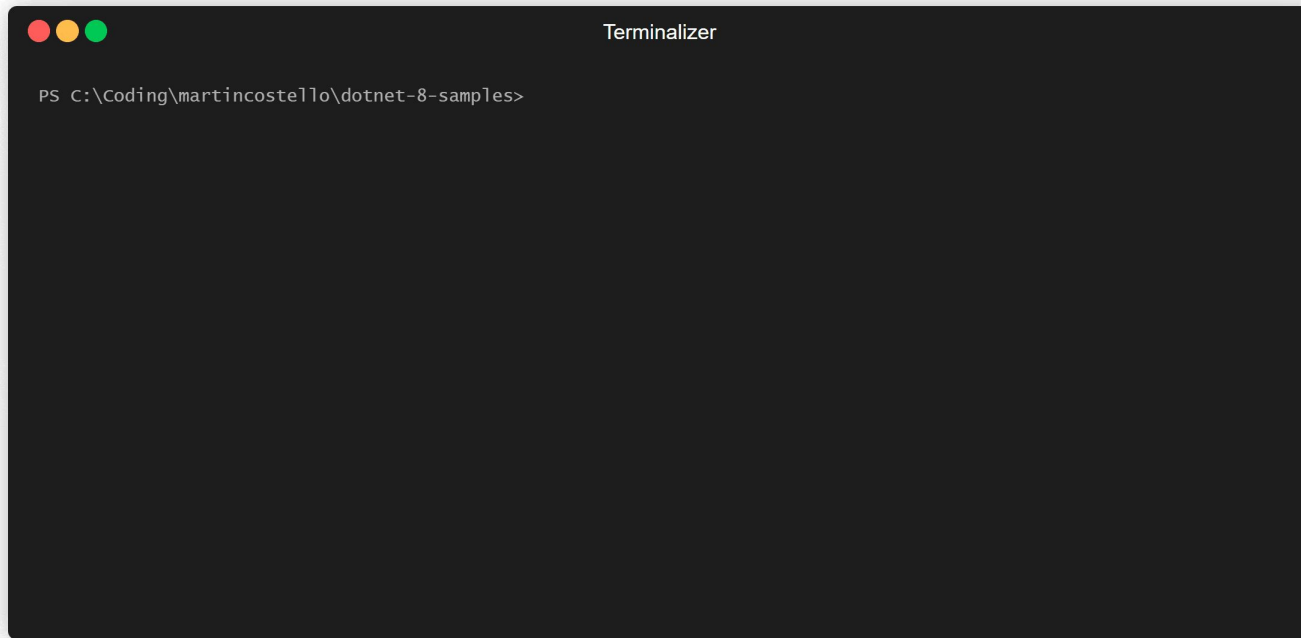
SDK features - Artifacts output



Benefits

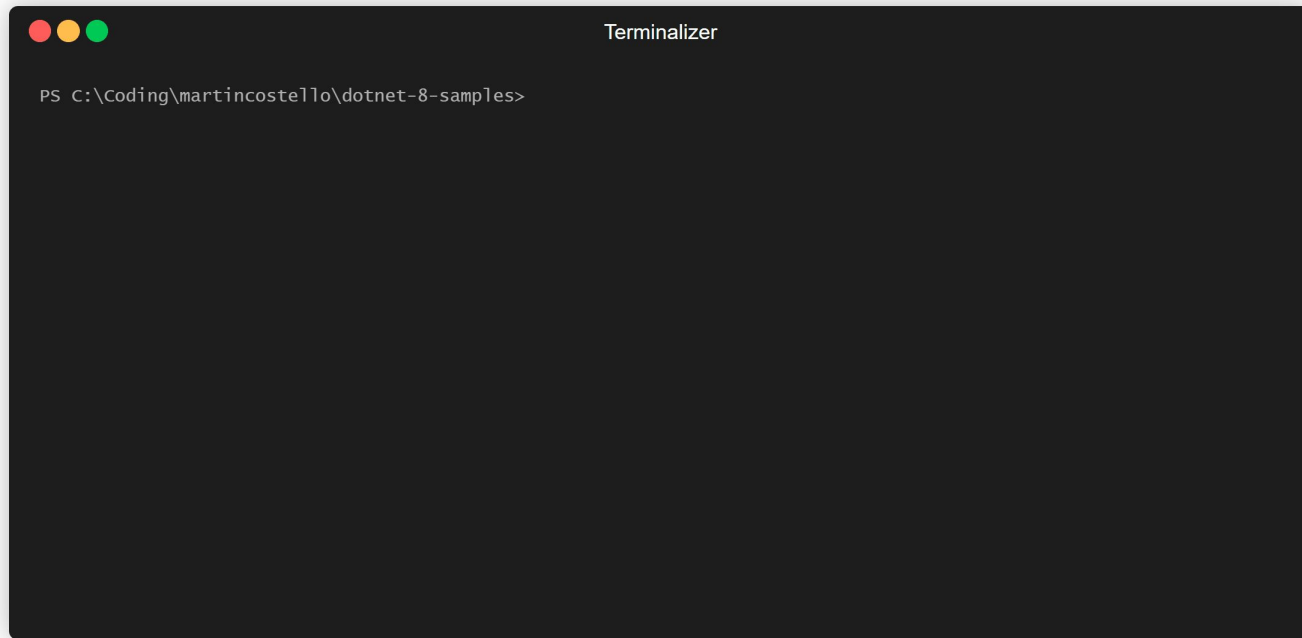
- Everything within on top-level folder
- Predictable locations
- Shorter paths when only one TFM
- Only one folder to  when disk space is low

SDK features 🧑 - Terminal logger 🪵



Without terminal logger 🧑 🪵

SDK features 🧑 - Terminal logger 🍷



Without terminal logger 🧑 🍷

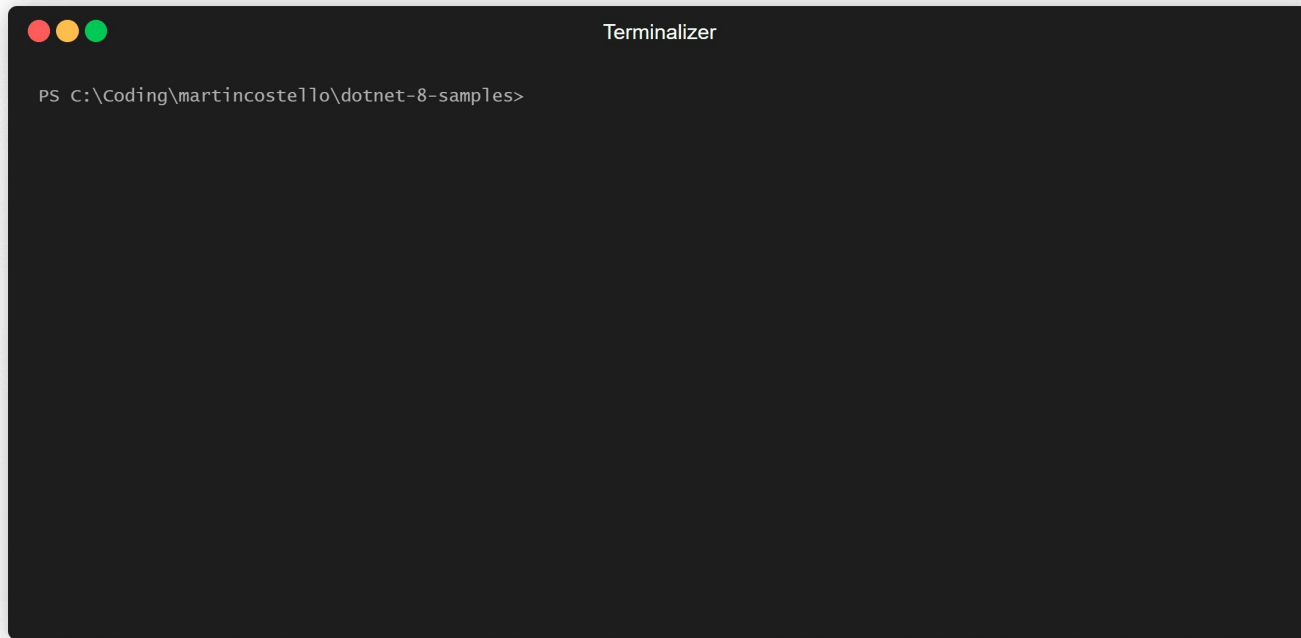
SDK features 🧑 - Terminal logger 🪵



[See here](#)

dotnet invoked with either --tl or MSBUILDTERMINALLOGGER=auto environment variable

SDK features 🧑 - Terminal logger 🪵



dotnet invoked with either `--tl` or `MSBUILDTERMINALLOGGER=auto` environment variable



Resources



- Microsoft Learn
 - [What's new in .NET 8](#)
 - [Breaking changes](#)
 - [Resilience libraries](#)
- [.NET Conf 2023](#)
- [This talk's samples](#)
- My GitHub - [@martincostello](#)
- My Twitter - [@martin_costello](#)



These
slides



Thank you! 🙏

@martin_costello



@martincostello

