

How we chose the right UI framework for DevToys 2.0

Etienne Baudoux | SDE, Microsoft



Multiple types of apps

Native

- Built for one platform using machine code
- Use native UI components
- Minimal customization
- Best performance
- Best integration

Cross Platform

- Single codebase for multiple platform
- May or may not use native UI components
- May or may not be compiled into platform-specific machine code

Hybrid

- Native business logic
- Web-based UI
- Easier to build and maintain
- Usually don't perform as fast as native apps

PWA

- Web apps designed to work offline and install like native apps
- Built with standard web technologies
- Don't have full access to device features
- Runs in a web browser

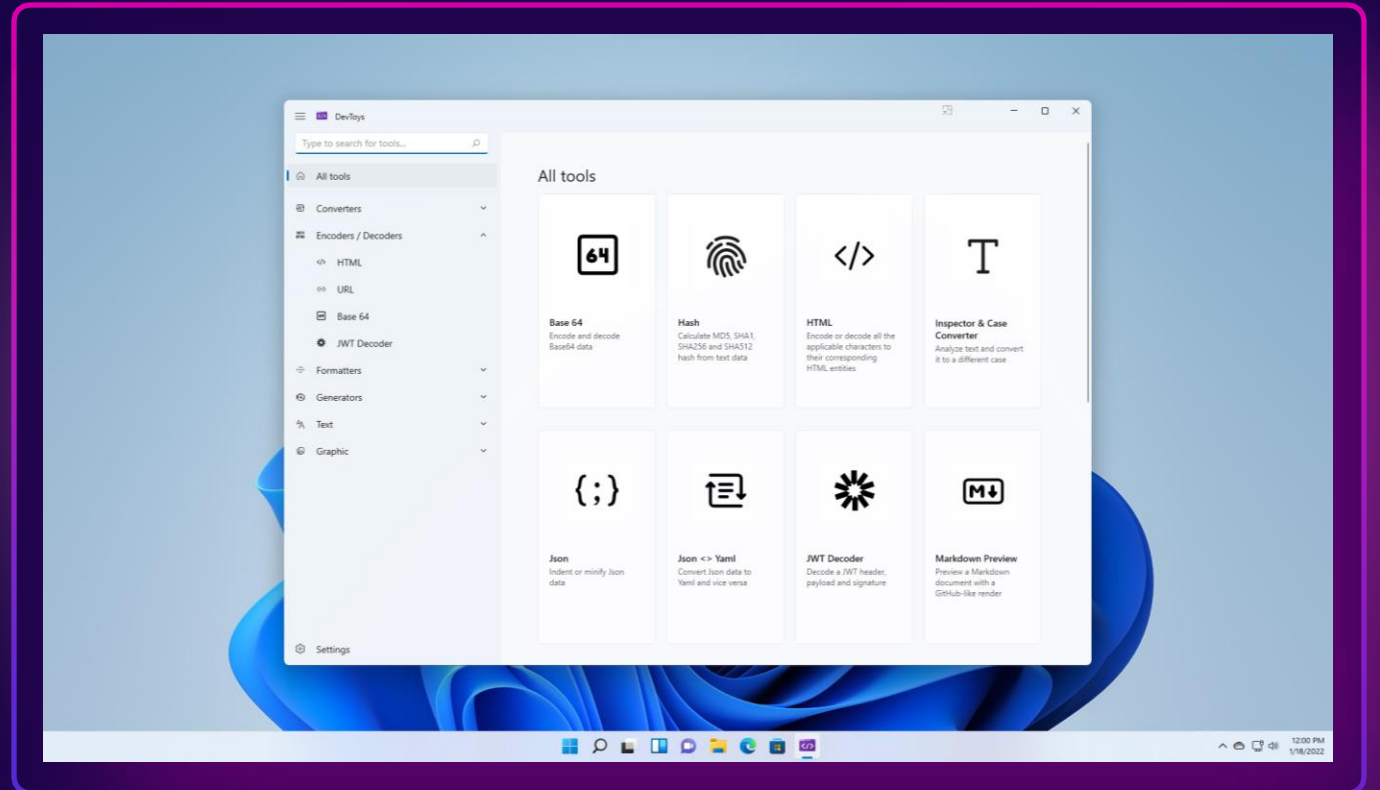
DevToys

From 1.0 to 2.0

DevToys

Swiss Army Knife for Developers

- Developed by
 - Etienne Baudoux
 - Benjamin Titeux
 - **NOT** by Microsoft
- Free and Open Source Software (FOSS)
- UWP app (native UI)
- devtoys.app




DevToys 1.0

What people liked

- Native-looking UI & Accessible
 - Fluent UI
- Windows integration
 - Smart Detection (clipboard)
 - TaskBar's JumpList
 - Pin tool to Start Menu
 - Multiple app instance
- Fast startup (~200ms)
- Works fully offline
- Monaco Editor

What people disliked

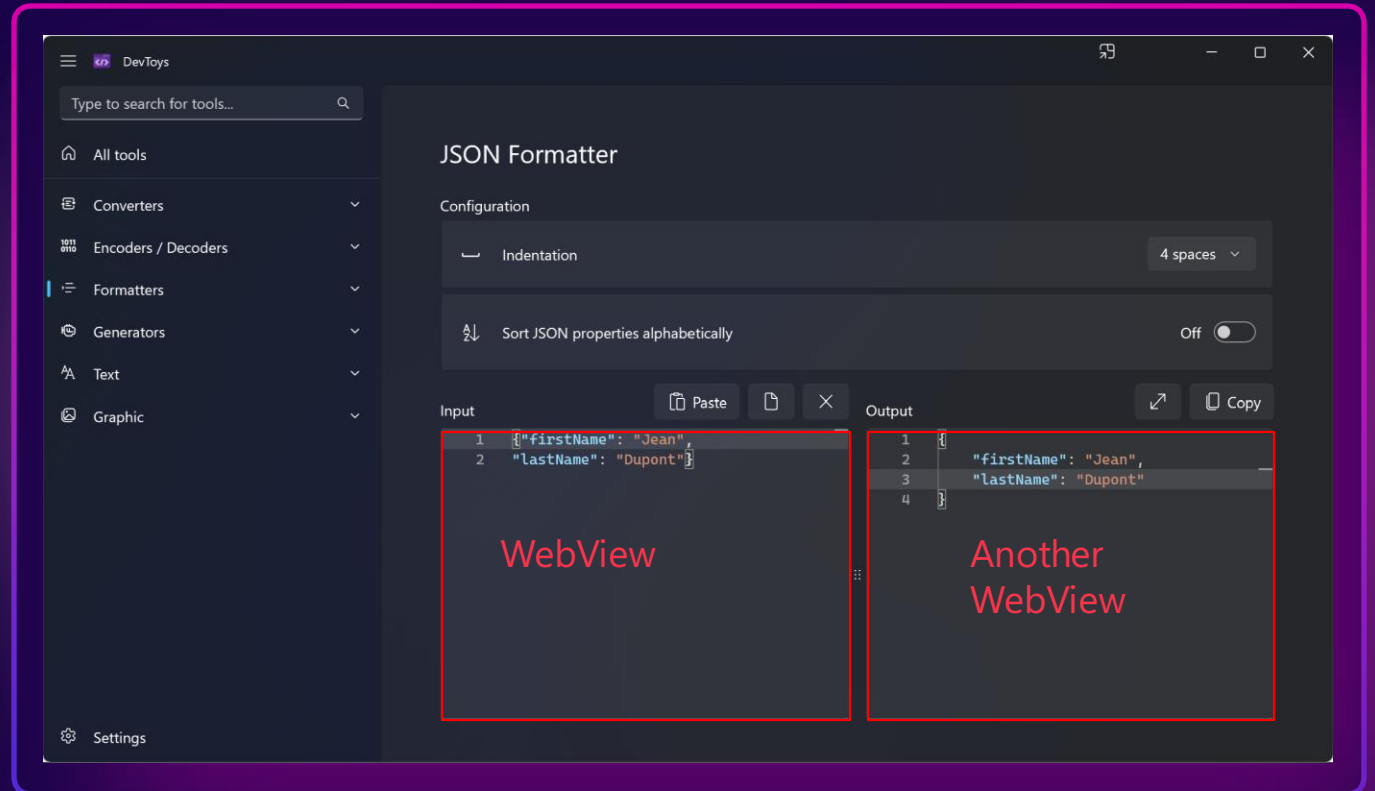
- Consumes a lot of RAM

^		31%	65%
Name	Status	CPU	Memory
Apps (8)			
>  DevToys (2)	Efficiency mode	0%	822.3 MB

DevToys 1.0

Monaco Editor

- microsoft.github.io/monaco-editor
- FOSS web-based code editor module
- Same editor than Visual Studio Code
- Runs in WebView



Most upvoted issue on GitHub

Suggestion: Cross-platform UI (Windows, MacOS, Linux) #156

✓ Closed

AmiK2001 opened this issue on Jan 19, 2022 · 65 comments



AmiK2001 commented on Jan 19, 2022 · edited ▾

Is your feature request related to a problem? Please describe.

The application is cool, but so far it only supports Windows OS.

Describe the solution you'd like

[AvaloniaUI](#) is a cross-platform, open source, XAML-based framework that is close to WPF.

It would be nice if a GUI based on AvaloniaUI became optional or even core to make it easier to support a cross-platform user interface.

Describe alternatives you've considered

As a cross-platform alternative to AvaloniaUI, I see only [Uno Platform](#).



164



1



35



20

From DevToys 1.0 to 2.0

- Requirements
 - Run on Windows, macOS and Linux
 - Maintain native-looking UI on Windows
 - And more (extensibility...)
- Nice to have
 - Keep Monaco Editor as code editor



Native?

First consideration



First consideration: Native?

Why trying native?

- Users do not like waiting. Native app offer faster startup and reaction to interaction
- The app UI will be consistent with the look and feel of the target OS
 - .NET Workloads allow to create native UI without any framework on top of it

Why not trying native?

- We are only 2 people working on it
- Our background is mainly Windows
 - We would need to learn UI API and paradigm for each platforms
- Learning curve is high
- Development and maintenance cost is too high
 - No financial cost in our case

So... let's try cross-platform?

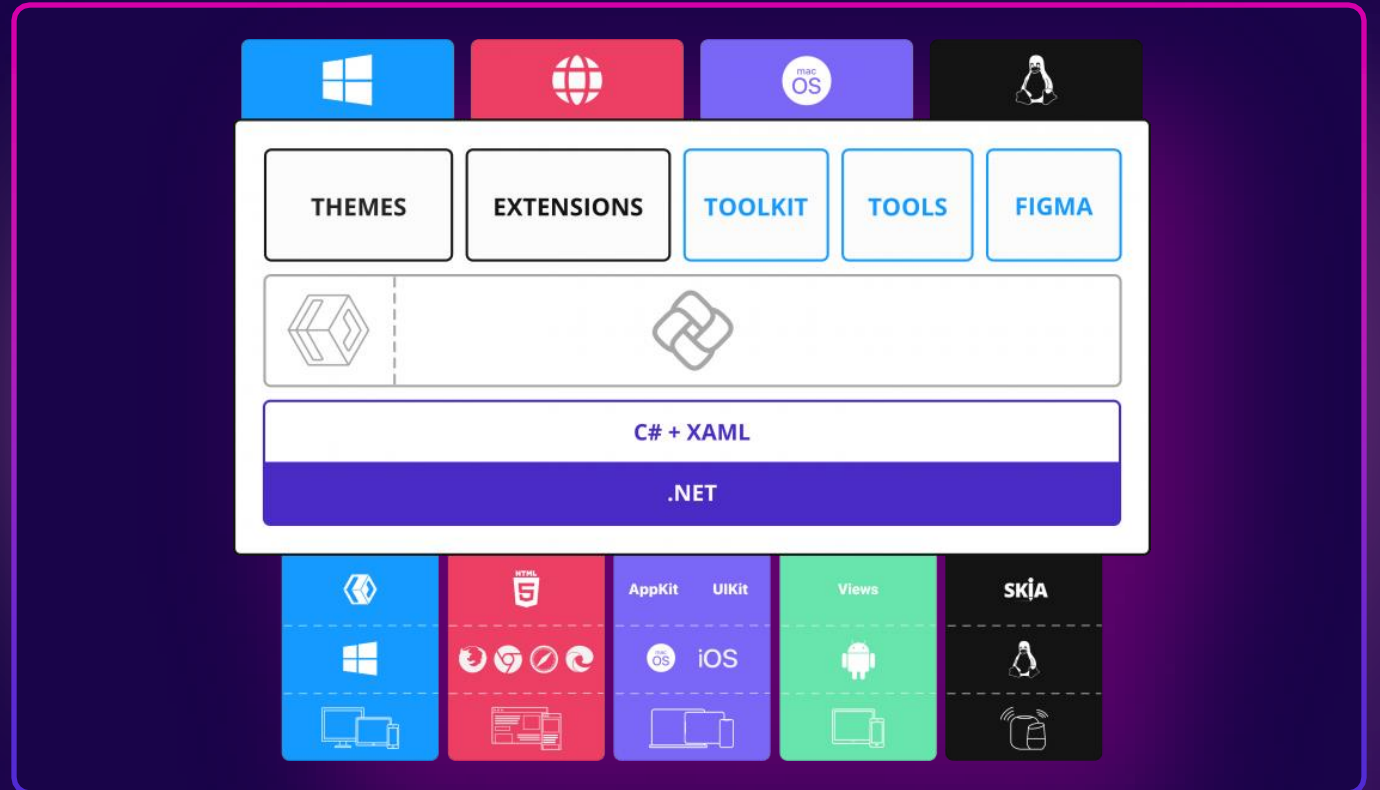
Uno Platform

First attempt

Uno Platform

What is it?

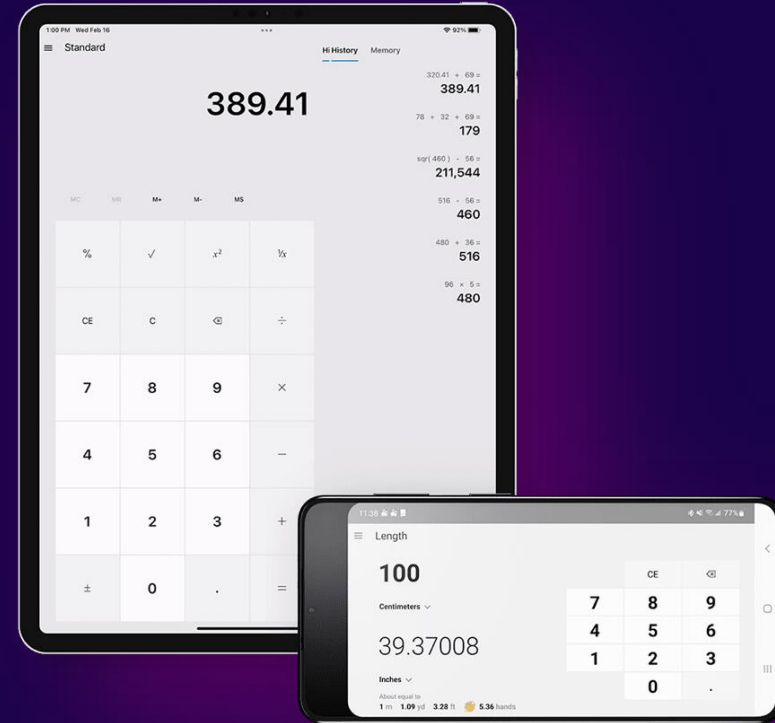
- FOSS
- One code, all platforms
 - Windows 7 / 8 / 10 / 11, Linux, macOS, iOS, Android, WebAssembly
 - Uses native UI components when possible
- platform.uno



Uno Platform

Why trying it for DevToys?

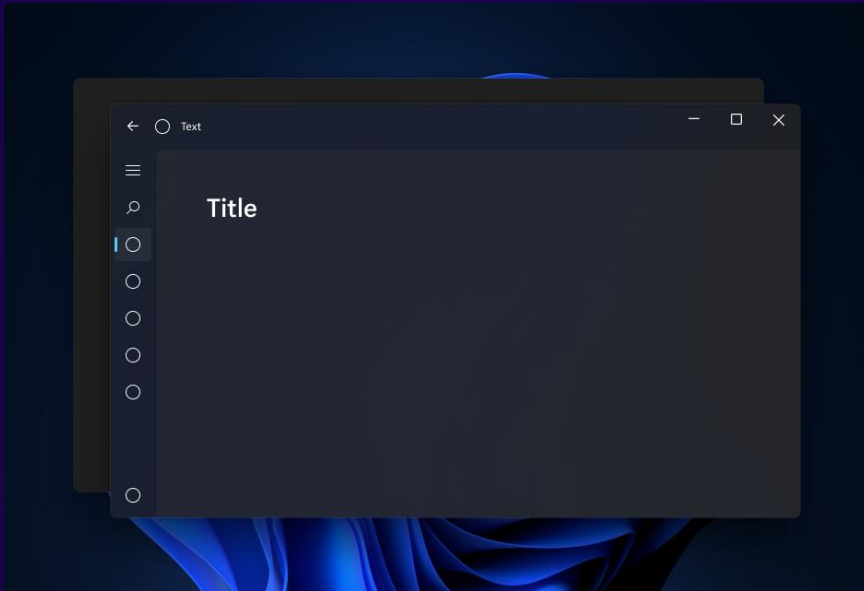
- Matches the same XAML than in UWP
- Windows
 - Render UI using WinUI 3
- macOS
 - Relies on Mac Catalyst
 - At the time of our test, MacOS support was not good. It has improved since then.
- Linux
 - Render UI using SKIA (2D graphic library)



Problems we encountered

Windows

- WinUI 3's WebView does not support transparency
 - No Mica support



macOS

- Bug in WebView in Mac Catalyst preventing keyboard input
 - Also impacts apps made in Swift
 - Bug still not fixed by Apple
developer.apple.com/forums/thread/721141

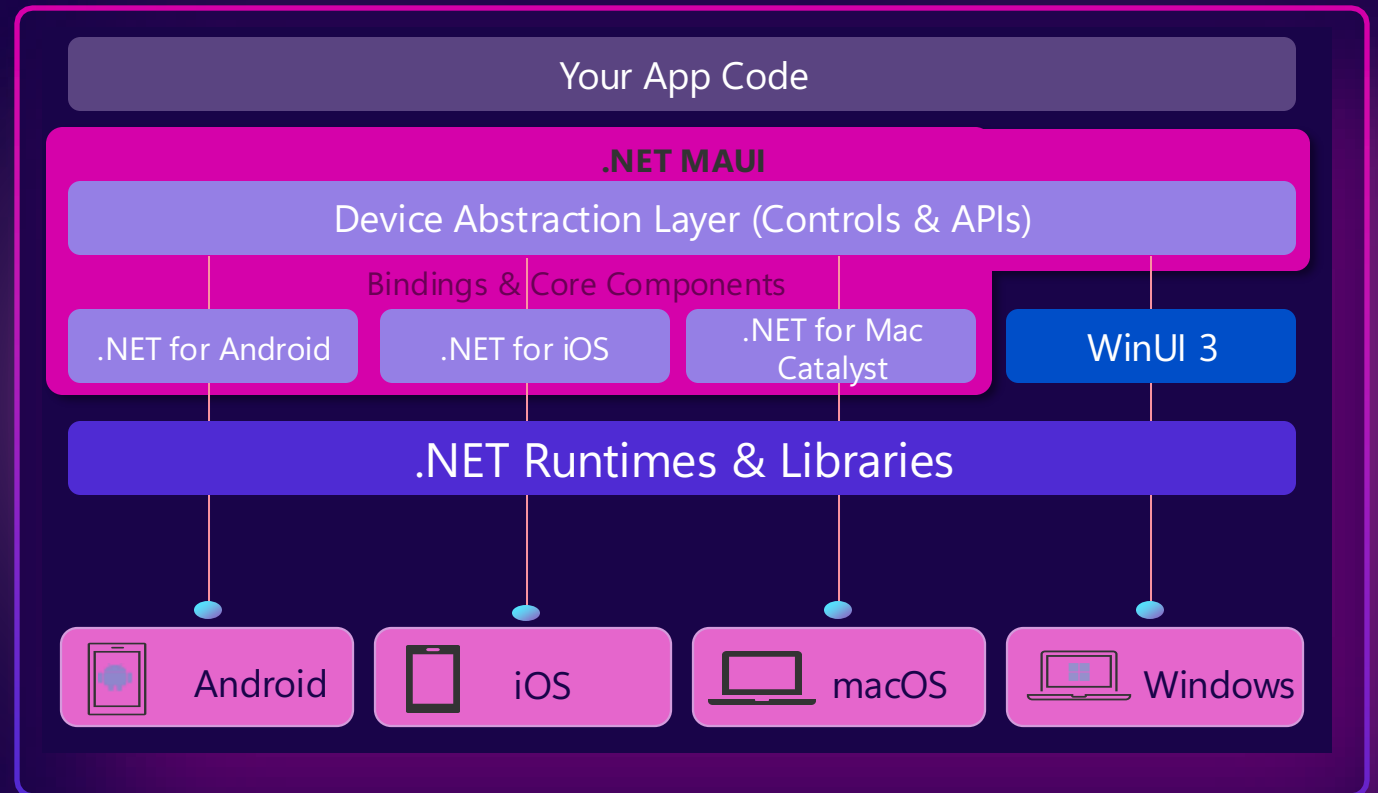
.NET MAUI

Second attempt

.NET MAUI

What is it?

- FOSS
- Evolution of Xamarin.Forms
- Targets Windows & Smartphone primarily
 - Windows 10 / 11, iOS, Android, macOS through Mac Catalyst
 - Uses native UI components
- dotnet.microsoft.com/apps/maui



Problems we encountered

Windows & macOS

- Same as



Linux

- Not supported



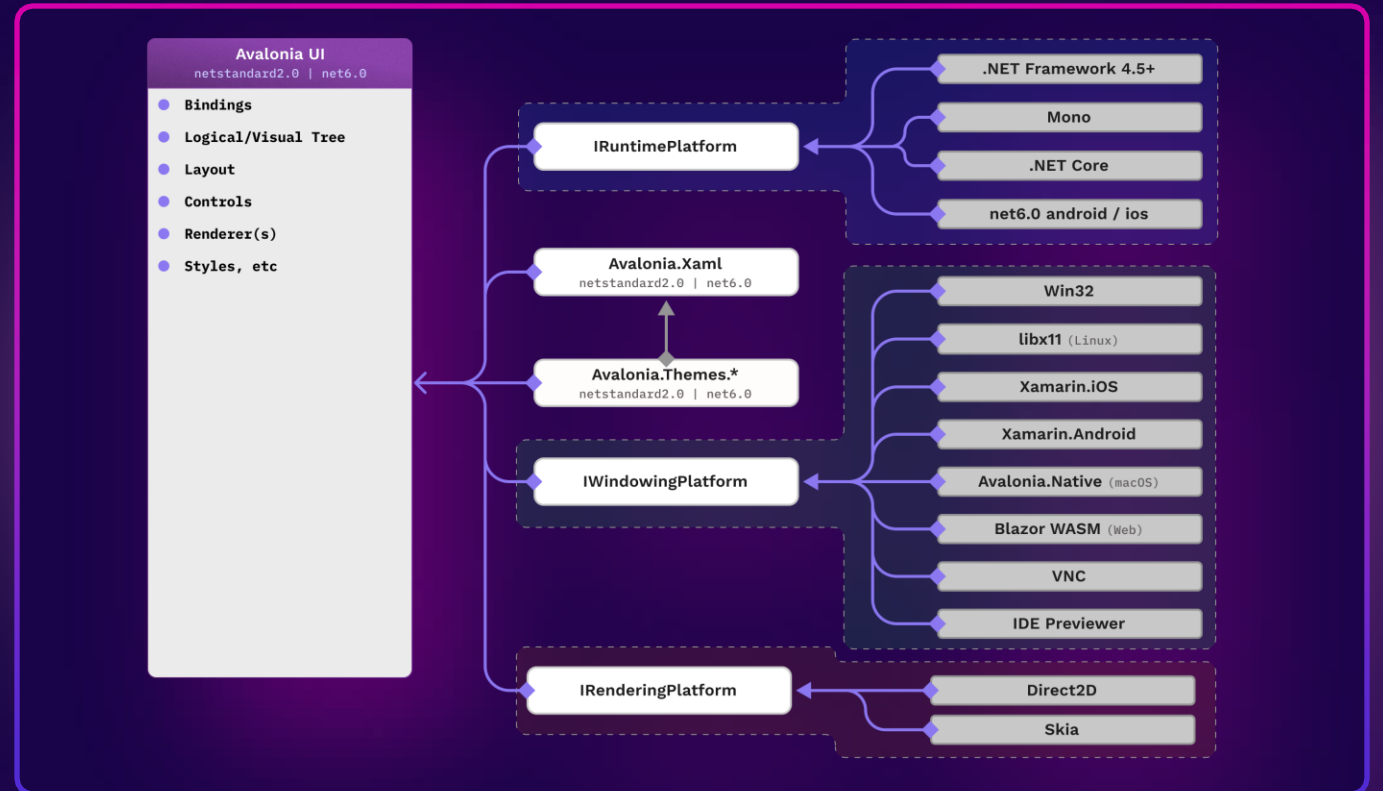
Avalonia UI

Third attempt

Avalonia UI

What is it?

- Unlike Uno and MAUI, specialized in desktop development, until recently
 - Windows 7 / 8 / 10 / 11, MacOS, Linux, iOS, Android, WebAssembly
- FOSS
- avaloniaui.net



Problems we encountered

Windows

- No native looking UI on Windows by default
 - Third party library “FluentAvalonia” but not perfect back in 2023
- Window customization to support Mica effect was tricky

All platforms

- Lack of official support of WebView
 - Possible through third party library since Summer 2023



We just lost 4 months...

So... let's try hybrid?

Blazor Hybrid

Fourth attempt

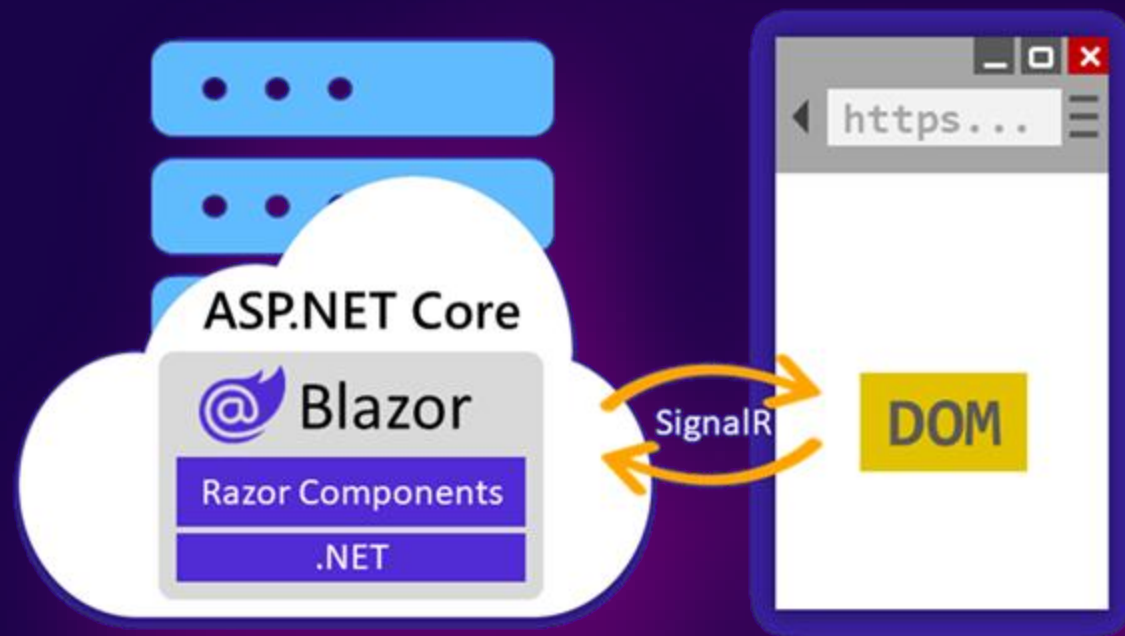


Blazor Hybrid

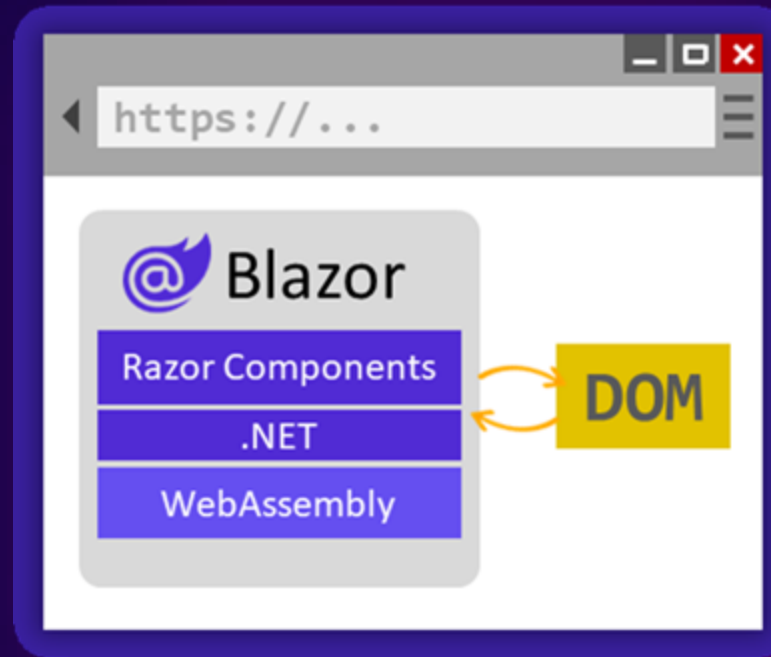
What is Blazor?

- Framework developed by Microsoft that allows developers to build interactive web applications using C# and .NET, running client-side or server-side.
- Blazor builds on the foundations of ASP.NET but offers new capabilities and a different development paradigm.
- 3 options available
 - Blazor Server / WebApp
 - Blazor WebAssembly
 - Blazor Hybrid

Blazor Server



Blazor WebAssembly



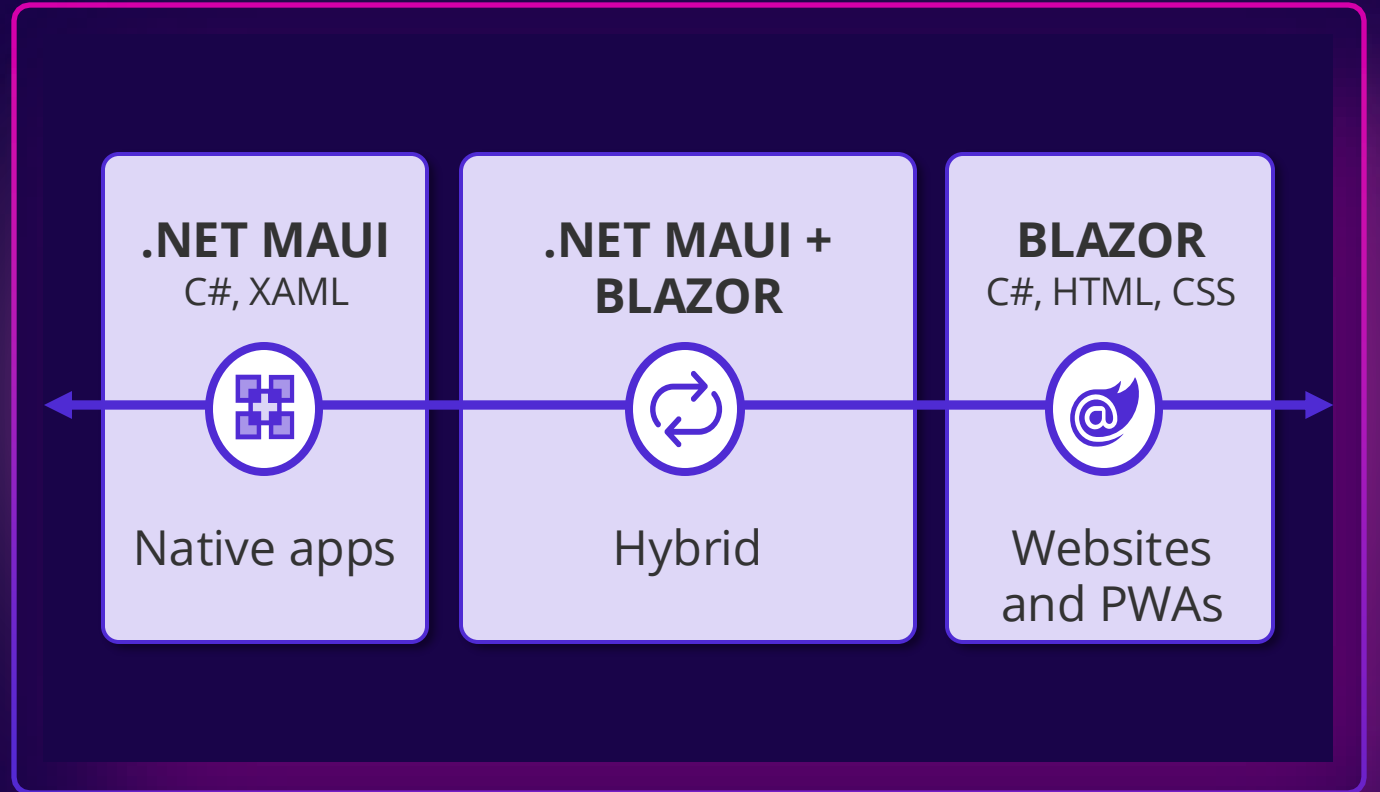
Hybrid apps with .NET & Blazor



Blazor Hybrid

BlazorWebView

- BlazorWebView officially supported by:
 - WinUI 3
 - WPF
 - WinForms
 - .NET MAUI

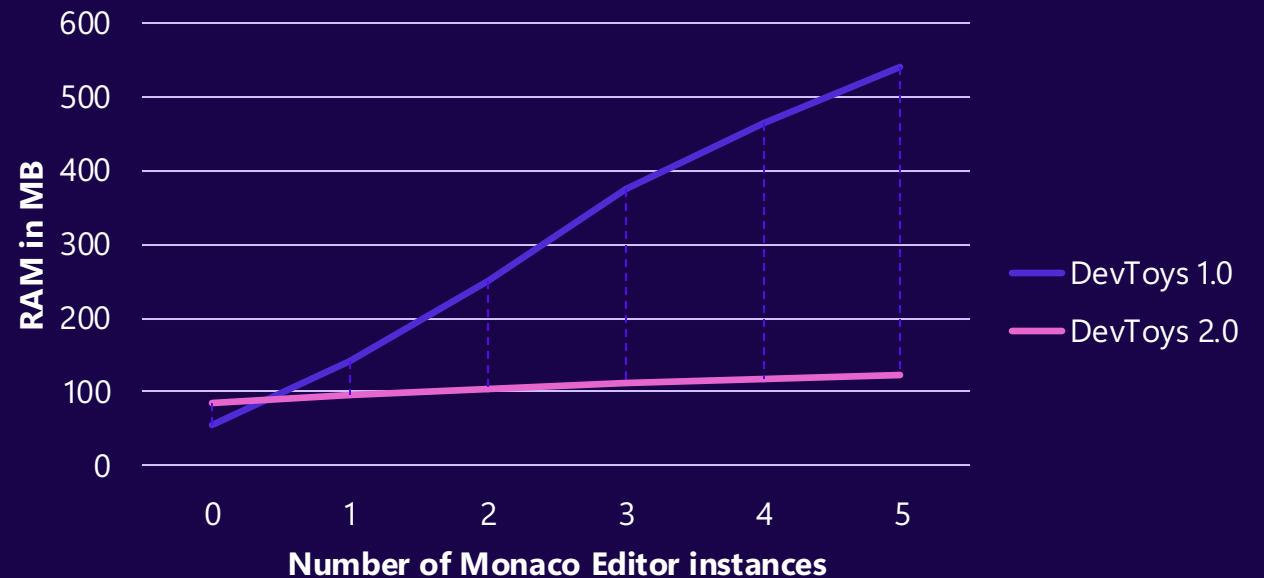


Blazor Hybrid

Why trying it for DevToys?

- A single Web View: solves the high memory consumption issue
- Similar to Electron app, but different
 - No web-browser shipped with the app
 - Chromium on Windows (WebView2) & Android
 - WebKit on iOS and MacOS
 - Pro: Smaller package than Electron app
 - Cons: Engine varies per OS

DevToys RAM consumption



Our solution

Our approach, based on learnings

Windows

- WinUI 3's WebView does not support transparency, hence, no Mica effect
- Workaround:
 - Use WPF
 - + WPF's BlazorWebView
 - + Win32 API (for Mica)

macOS

- Mac Catalyst (UIKit) is out of question due to bug with keyboard input
- Workaround:
 - .NET macOS app (AppKit)
 - Fork of .NET MAUI's Blazor WebView

Linux

- .NET app with GTK
- Fork of .NET MAUI's Blazor WebView



**How to make the UI looking
native?**

maui

Home

Counter

Fetch data

About

Weather forecast

This component demonstrates fetching data from local disk.

Date	Temp. (C)	Temp. (F)	Summary
8/24/2021	45	112	Scorching
8/25/2021	8	46	Bracing
8/26/2021	18	64	Warm
8/27/2021	-17	2	Sweltering
8/28/2021	35	94	Bracing

Mud Blazor

FOSS

www.mudblazor.com

Radzen

Partially FOSS

blazor.radzen.com

Fluent UI Blazor

FOSS

Fluent UI for Web - Not Desktop

www.fluentui-blazor.net

Fluent Svelte

FOSS

Looks perfectly like Windows 11's

Fluent UI

fluent-svelte.vercel.app

Bonus: Blazorise, DevExpress, Telerik, Syncfusion and many more...



Type to search for tools...



All tools



Converters



Encoders / Decoders



Formatters



Generators



Graphic



Testers



Text



Support the development



Manage extensions



Settings

Welcome to DevToys v2.0.0-pre.0



Base64 Image Encoder / Decoder
Encode and decode Base64 image data



Base64 Text Encoder / Decoder
Encode and decode Base64 text data



Certificate Decoder
Decode a certificate



Color Blindness Simulator
Simulate color blindness on a picture or screenshot

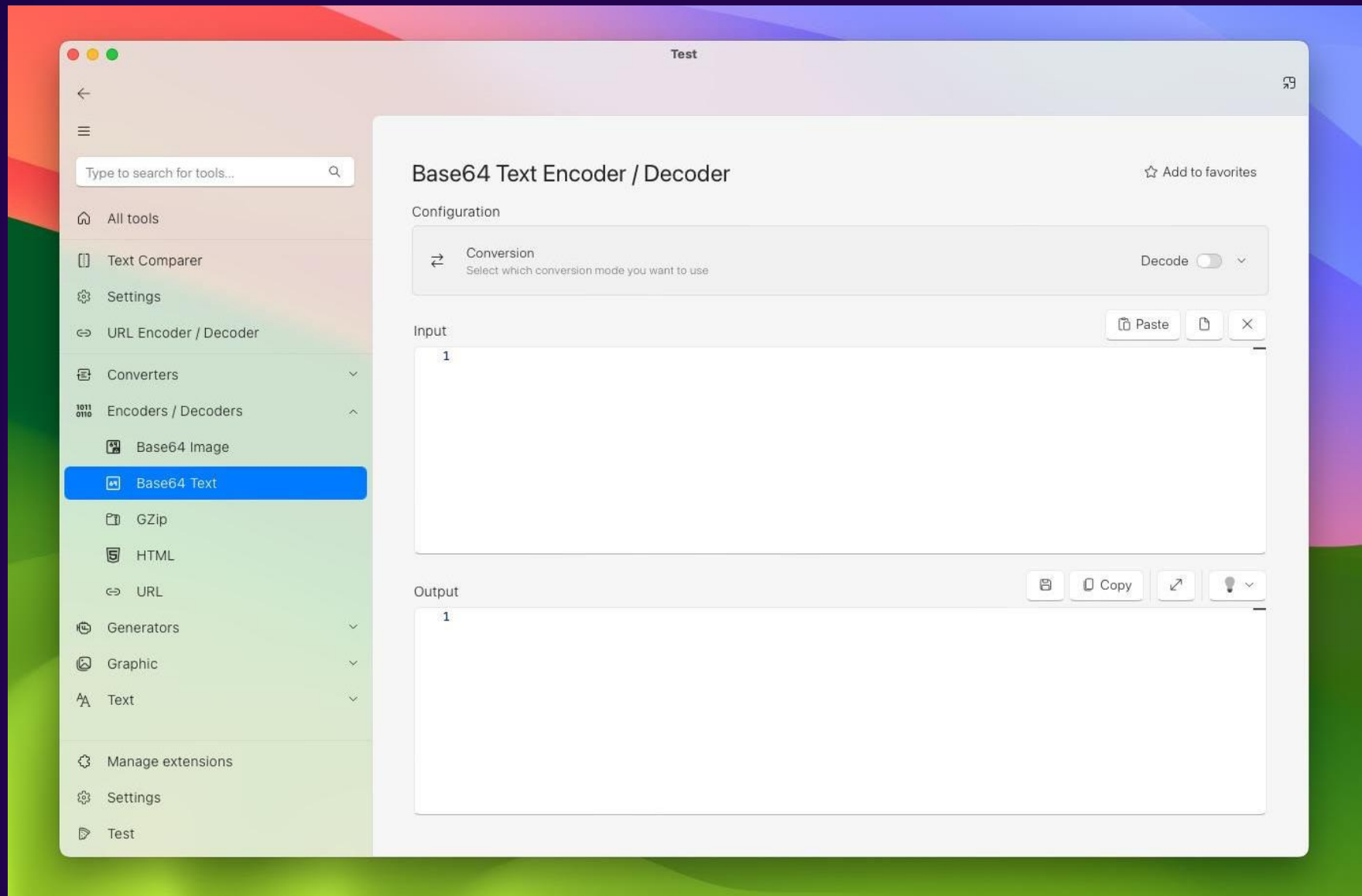


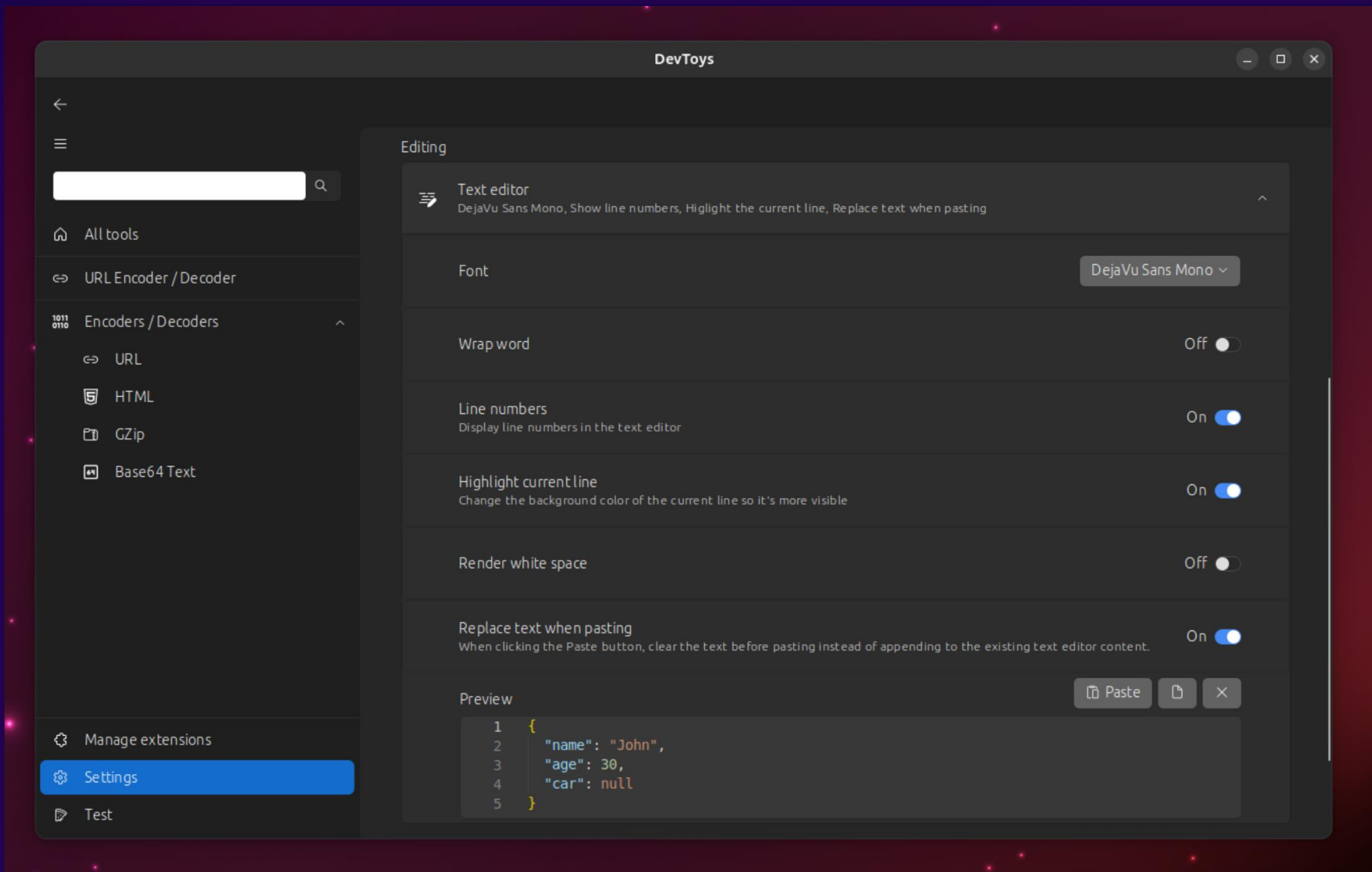
Cron expression parser
Parse Cron expression to get scheduled dates



Date Converter
Convert date to human-readable date and vice versa







Conclusion



**Choosing the right .NET UI framework is a
unique journey for every project.**

Some advice...

Budget

If high, consider native.
If low, consider PWA.

Performance

If important, consider native.
If not, consider Hybrid or PWA.

Accessibility

If important, consider native UI.

Skills

If your team is made of web dev,
consider Hybrid or PWA.

Targeted Platforms

Windows only? WinUI 3
Smartphone? .NET MAUI
Everything? Uno Platform or
Avalonia UI

More...

Thank you

