

A portrait of a young man with short, light-colored hair, looking slightly to his right with a neutral expression. He is wearing a dark, ribbed t-shirt. The background is dark and out of focus.

Контур

ЧТО ТАМ С BLAZOR WEBASSEMBLY?

Рома Неволин

DevRel в Контур

Announcing .NET 6 — The Fastest .NET Yet



Richard

November 8th, 2021 | 77

Welcome to .NET 6. Today's release is the result of just over a [year's worth of effort](#) by the .NET Team and community. C# 10 and F# 6 deliver language improvements that make your code simpler and better. There are [massive gains in performance](#), which we've seen dropping the cost of hosting cloud services at Microsoft. .NET 6 is the first release that natively supports Apple Silicon (Arm64) and has also been improved for Windows Arm64. We built a new dynamic profile-guided optimization (PGO) system that delivers deep optimizations that are only possible at runtime. Cloud diagnostics have been improved with [dotnet monitor](#) and [OpenTelemetry](#). [WebAssembly](#) support is more capable and performant. New APIs have been added, for [HTTP/3](#), [processing JSON](#), [mathematics](#), and directly manipulating memory. .NET 6 will be [supported for three years](#). Developers have already started upgrading applications to .NET 6 and we've heard great early results in production. .NET 6 is ready for your app.

You can [download .NET 6](#) for Linux, macOS, and Windows.

- [Installers and binaries](#)
- [Container images](#)
- [Linux packages](#)
- [Release notes](#)
- [API diff](#)
- [Known issues](#)
- [GitHub issue tracker](#)

See the [ASP.NET Core](#), [Entity Framework](#), [Windows Forms](#), [.NET MAUI](#), [YARP](#), and [dotnet monitor](#) posts for what's new in a variety of scenarios.

Visual Studio 2022 is also releasing today. [Read the announcement](#) and [watch the launch event](#) to learn more about it.

Announcing .NET 6 — The Fastest .NET Yet



Richard

November 8th, 2021 | 77

Welcome to .NET 6. Today's release is the result of just over a [year's worth of effort](#) by the .NET Team and community. C# 10 and F# 6 deliver language improvements that make your code simpler and better. There are [massive gains in performance](#), which we've seen dropping the cost of hosting cloud services at Microsoft. .NET 6 is the first release that natively supports Apple Silicon (Arm64) and has also been improved for Windows Arm64. We built a new dynamic profile-guided optimization (PGO) system that delivers deep optimizations that are only possible at runtime. Cloud diagnostics have been improved with [dotnet monitor](#) and [OpenTelemetry](#). [WebAssembly](#) support is more capable and [performant](#). New APIs have been added, for [HTTP/3](#), [processing JSON](#), [mathematics](#), and directly manipulating memory. .NET 6 will be [supported for three years](#). Developers have already started upgrading applications to .NET 6 and we've heard great early results in production. .NET 6 is ready for your app.

You can [download .NET 6](#) for Linux, macOS, and Windows.

- [Installers and binaries](#)
- [Container images](#)
- [Linux packages](#)
- [Release notes](#)
- [API diff](#)
- [Known issues](#)
- [GitHub issue tracker](#)

See the [ASP.NET Core](#), [Entity Framework](#), [Windows Forms](#), [.NET MAUI](#), [YARP](#), and [dotnet monitor](#) posts for what's new in a variety of scenarios.

Visual Studio 2022 is also releasing today. [Read the announcement](#) and [watch the launch event](#) to learn more about it.

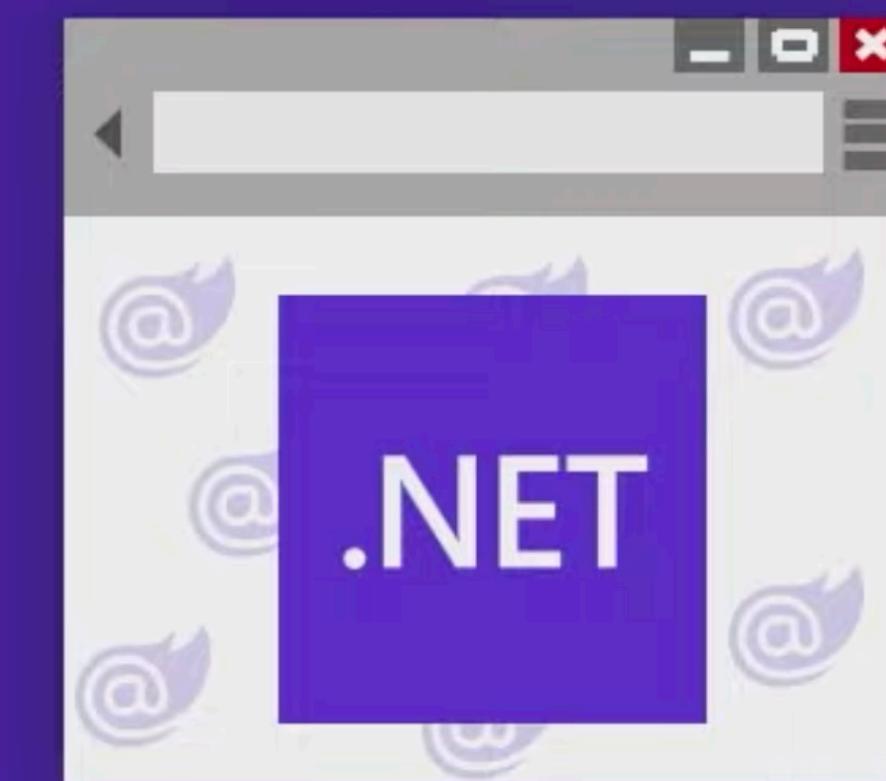
- Blazor [components can now be rendered from JavaScript](#) and integrated with existing JavaScript based apps.
- WebAssembly AOT [compilation for Blazor WebAssembly \(Wasm\) apps](#), as well as support for runtime relinking and native dependencies.

ТАК ЭТИМ МОЖНО ПОЛЬЗОВАТЬСЯ?

Modern web UI with .NET & Blazor



Server



WebAssembly



Hybrid

ПОЧЕМУ БОЯТСЯ WASM?

- А вдруг браузер не поддерживает?
- А что там по перфомансу?
- А как это совместимо с остальной экосистемой?

	Your browser	 Chrome ⁹⁶	 Firefox ⁹⁰	 Safari ^{15.2}	 Wasmtime ^{0.33}	 Wasmer ^{2.0}
Standardized features						
JS BigInt to Wasm i64 integration	✓	✓	✓	✓	✗/a	✗/a
Bulk memory operations	✓	✓	✓	✓	✓	✓
Multi-value	✓	✓	✓	✓	✓	✓
Import & export of mutable globals	✓	✓	✓	✓	✓	✓
Reference types	✓	✓	✓	✓	✓	✓
Non-trapping float-to-int conversions	✓	✓	✓	✓	✓	✓
Sign-extension operations	✓	✓	✓	✓	✓	✓
Fixed-width SIMD	✓	✓	✓	✗	✓	✓
In-progress proposals						
Exception handling	✓	✓	⏳	✓	✗	✗
Extended constant expressions	✗	✗	⏳	✗	✗	✗
Memory64	✗	✗	⏳	✗	⏳	✗
Multiple memories	✗	✗	✗	✗	⏳	✗
Module Linking	✗	✗	✗	✗	⏳	✗
Tail calls	✗	⏳	✗	✗	✗	✗
Threads and atomics	✓	✓	✓	✓	✗	⏳

ПОЧЕМУ БОЯТСЯ WASM?

- А вдруг браузер не поддерживает?
- А что там по перфомансу?
- А как это совместимо с остальной экосистемой?

Announcing .NET 6 — The Fastest .NET Yet



Richard

November 8th, 2021 | 77

Welcome to .NET 6. Today's release is the result of just over a [year's worth of effort](#) by the .NET Team and community. C# 10 and F# 6 deliver language improvements that make your code simpler and better. There are [massive gains in performance](#), which we've seen dropping the cost of hosting cloud services at Microsoft. .NET 6 is the first release that natively supports Apple Silicon (Arm64) and has also been improved for Windows Arm64. We built a new dynamic profile-guided optimization (PGO) system that delivers deep optimizations that are only possible at runtime. Cloud diagnostics have been improved with [dotnet monitor](#) and [OpenTelemetry](#). [WebAssembly](#) support is more capable and [performant](#). New APIs have been added, for [HTTP/3](#), [processing JSON](#), [mathematics](#), and directly manipulating memory. .NET 6 will be [supported for three years](#). Developers have already started upgrading applications to .NET 6 and we've heard great early results in production. .NET 6 is ready for your app.

You can [download .NET 6](#) for Linux, macOS, and Windows.

- [Installers and binaries](#)
- [Container images](#)
- [Linux packages](#)
- [Release notes](#)
- [API diff](#)
- [Known issues](#)
- [GitHub issue tracker](#)

See the [ASP.NET Core](#), [Entity Framework](#), [Windows Forms](#), [.NET MAUI](#), [YARP](#), and [dotnet monitor](#) posts for what's new in a variety of scenarios.

Visual Studio 2022 is also releasing today. [Read the announcement](#) and [watch the launch event](#) to learn more about it.

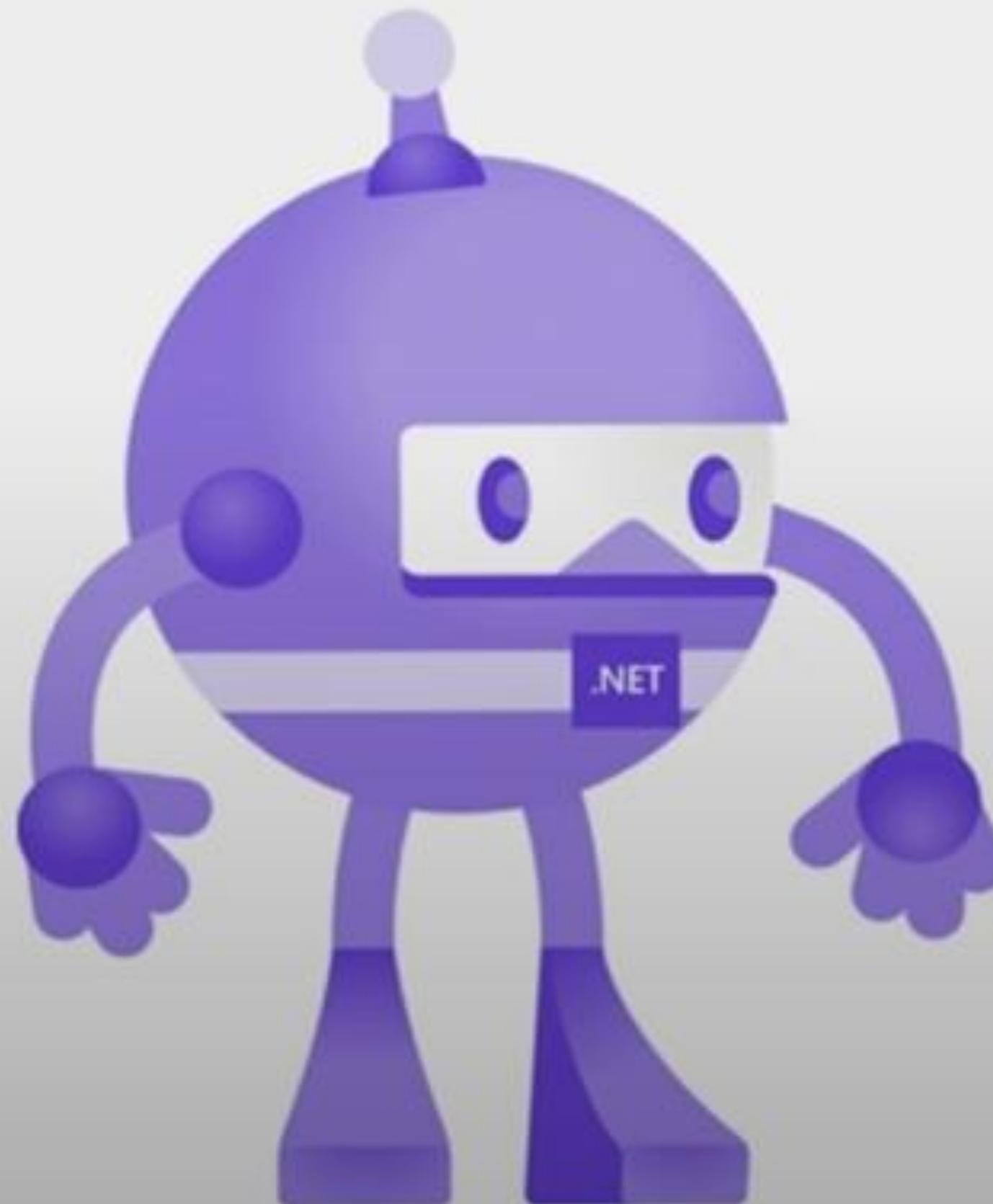
Компиляция AOT

Blazor WebAssembly поддерживает [компиляцию в режиме Ahead Of Time \(AOT\)](#), в котором код .NET можно скомпилировать непосредственно в WebAssembly. Компиляция AOT позволяет повысить производительность среды выполнения за счет увеличения размера приложения.

Blazor WebAssembly trimming

.NET 5

- Publish size: 1.7 MB



.NET 6

- Publish size: 1.0 MB
- **~40% size reduction**



БОЛЬШЕ — В БЛОГЕ РАЗРАБОТЧИКОВ

devblogs.microsoft.com/dotnet/performance-improvements-in-net-6/#blazor-and-mono

ПОЧЕМУ БОЯТСЯ WASM?

- А вдруг браузер не поддерживает?
- А что там по перфомансу?
- А как это совместимо с остальной экосистемой?

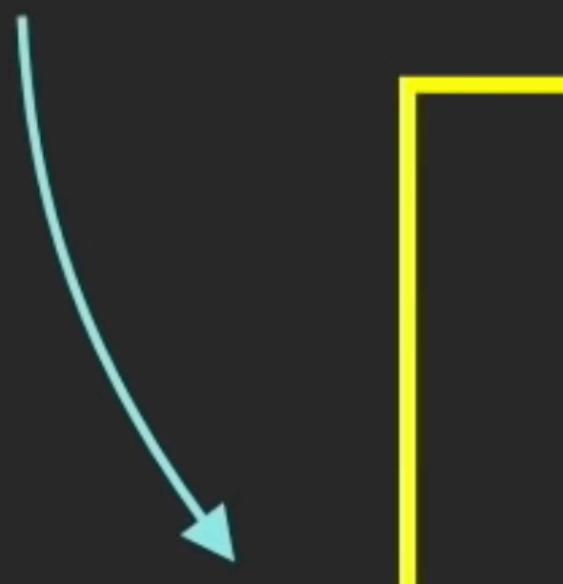


Central Servers



Need new inventory management UI

- fast, powerful querying
- no heavy server load
- keep working offline
- put it in all 3 apps
- use QR code lib from Rust team



Manufacturing Hub



Mission Control

Blazor custom elements

Experimental support is also now available for building custom elements with Blazor using the [Microsoft.AspNetCore.Components.CustomElements](#) NuGet package. Custom elements use [standard HTML interfaces](#) to implement custom HTML elements.

To create a custom element using Blazor, register a Blazor root component as custom elements like this:

```
options.RootComponents.RegisterAsCustomElement<Counter>("my-counter");
```

You can then use this custom element with any other web framework you'd like. For example, here's how you would use this Blazor counter custom element in a React app:

```
<my-counter increment-amount={incrementAmount}></my-counter>
```

```
@*Generate an Angular component*@
@attribute [GenerateAngular]
```

```
@*Generate an React component*@
@attribute [GenerateReact]
```

You then register the Blazor components as Angular or React components:

```
options.RootComponents.RegisterForAngular<Counter>();
```

```
options.RootComponents.RegisterForReact<Counter>();
```

When the project gets built, it generates Angular and React components based on your Blazor components. You then use the generated Angular and React components like you would normally:

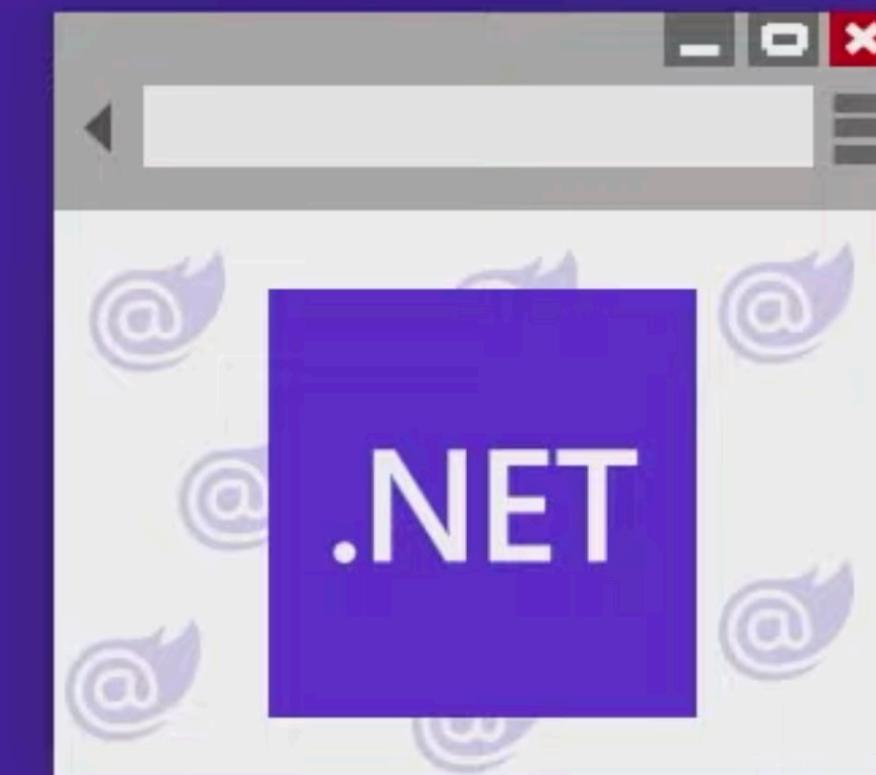
```
// Angular
<counter [incrementAmount]="incrementAmount"></counter>
```

```
// React
<Counter incrementAmount={incrementAmount}></Counter>
```

Modern web UI with .NET & Blazor



Server



WebAssembly



Hybrid

блазорвебвиев

Статья • 04.01.2022 • Чтение занимает 2 мин

Были ли сведения на этой странице полезными?  

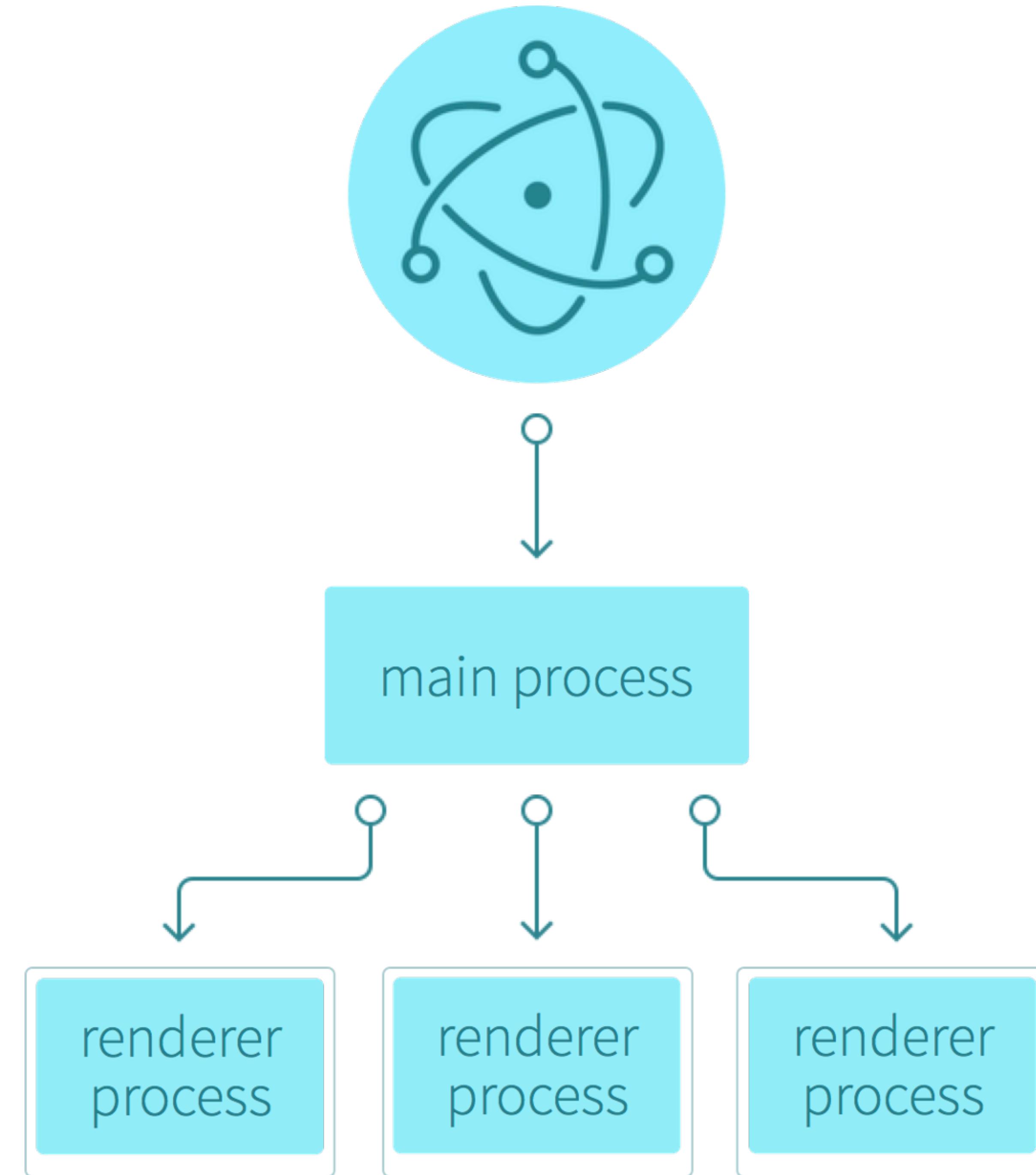
Пользовательский интерфейс многоплатформенного приложения .NET (.NET MAUI) `BlazorWebView` — это элемент управления, позволяющий размещать веб-приложение блазор в приложении .NET Мауи. Эти приложения,

известные как приложения .NET MAUI Блазор, позволяют интегрировать веб-приложение Блазор с функциями платформы и элементами управления пользовательского интерфейса. `BlazorWebView` Элемент управления может

быть добавлен на любую страницу приложения .NET Мауи и указывает на корень приложения блазор.

Компоненты Блазор работают изначально в процессе .NET и отображают веб-интерфейс в внедренном элементе управления веб-представления. Приложения .NET MAUI Блазор могут работать на всех plataформах, поддерживаемых .NET MAUI.

```
<b:BlazorWebView HostPage="wwwroot/index.html">
    <b:BlazorWebView.RootComponents>
        <b:RootComponent Selector="app" ComponentType="{x:Type local:Main}" />
    </b:BlazorWebView.RootComponents>
</b:BlazorWebView>
```



КОГДА Я МОГУ БРАТЬ BLAZOR WASM?

- Когда пишешь сложное браузерное приложение
- Когда поддержка старых браузеров не критична
- Когда готов немного рискнуть

КОГДА Я МОГУ БРАТЬ BLAZOR WASM?

- Когда пишешь сложное браузерное приложение
- Когда поддержка старых браузеров не критична
- Когда готов немного рискнуть

КОГДА Я МОГУ БРАТЬ BLAZOR WASM?

- Когда пишешь сложное браузерное приложение
- Когда поддержка старых браузеров не критична
- Когда готов немного рискнуть

Рома Неволин



nevoroman@gmail.com



t.me/nevoroman