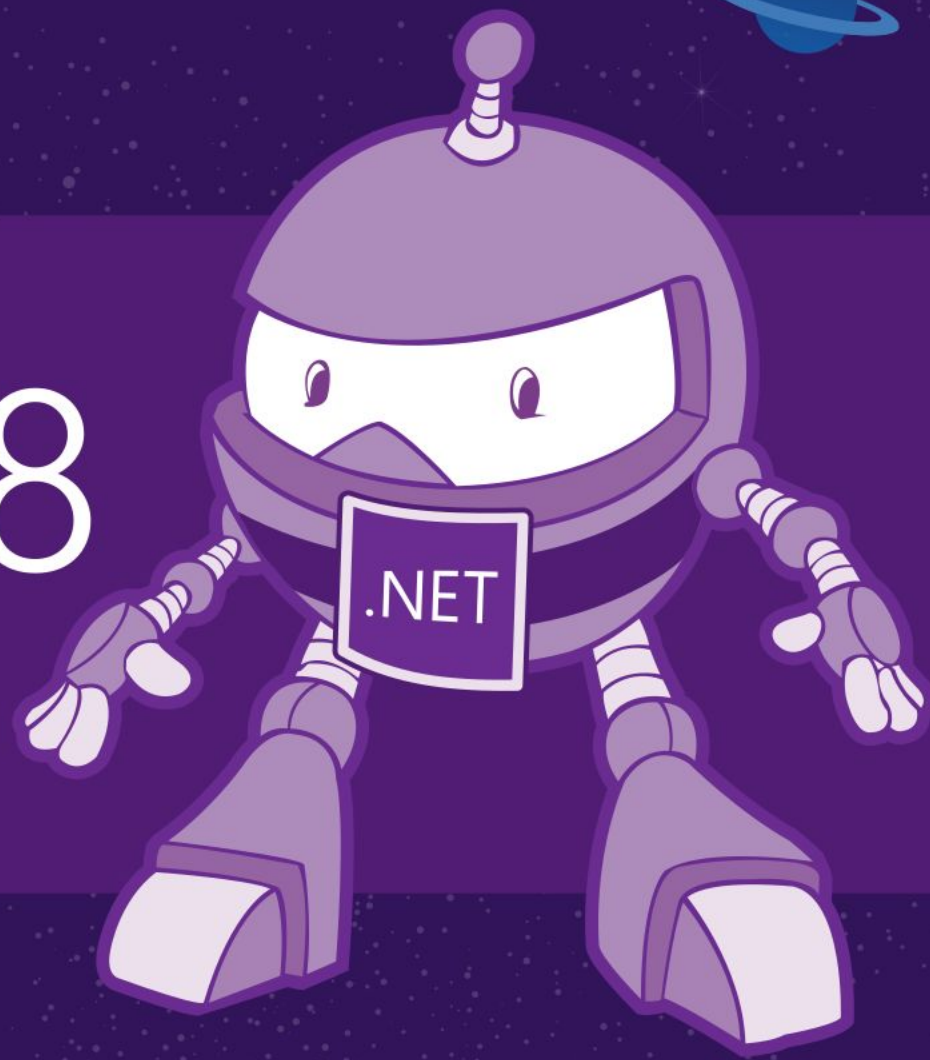


.NET Conf 2018

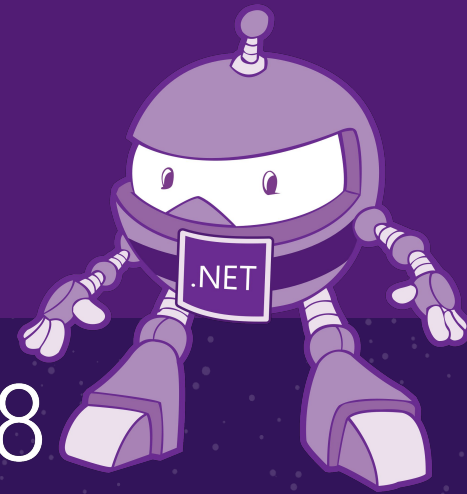
Discover the world of .NET



www.dotnetconf.net

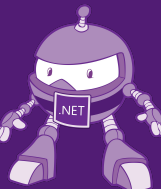
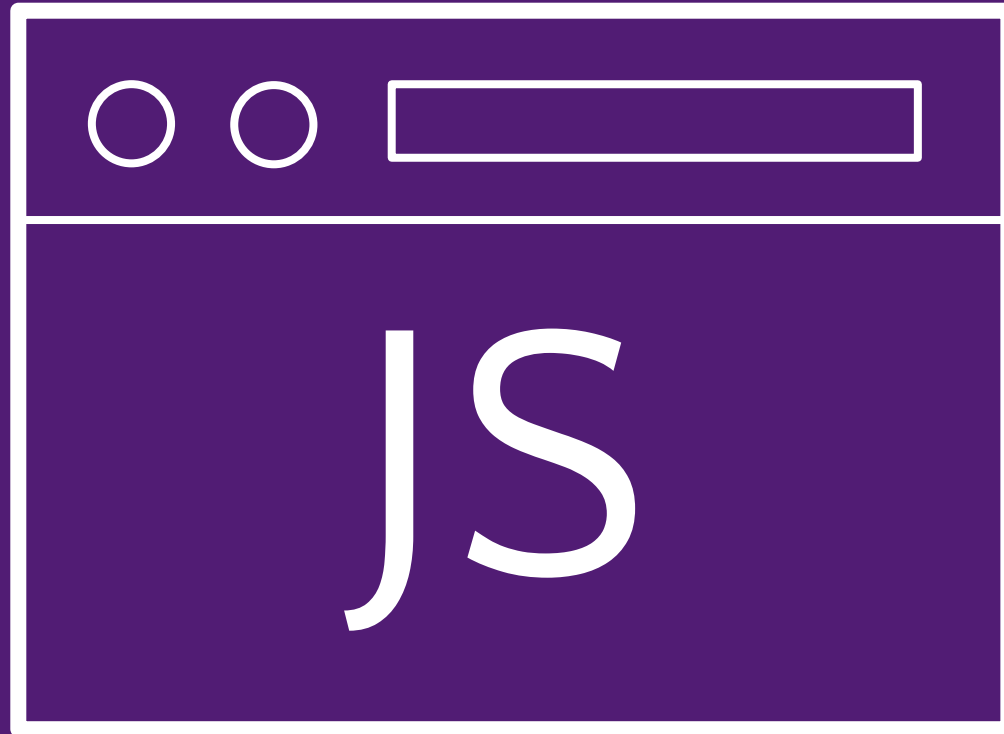
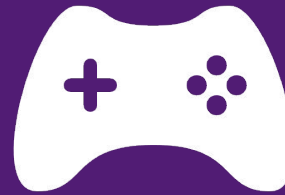
Blazor: Modern Web development with .NET and WebAssembly

Vorapat Nicklamai
Jetabroad



.NET Conf 2018

The slides are credited to Daniel Roth (Microsoft)



C#

Python

.NET

Go

F#

WA

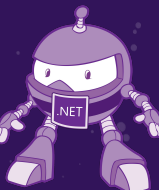
Ruby

Java

Rust

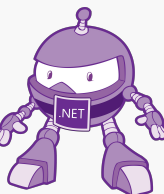
.wasm

An efficient, **low-level** bytecode
for the web



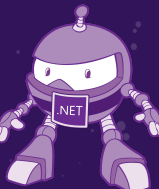
Why use .NET for browser apps?

- **Stable, mature, productive:** .NET Standard, MSBuild
- **Fast, scalable, reliable:** .NET Core for backend services
- **Modern languages:** Innovations in C#, F#, Razor
- **First-rate dev tools:** Visual Studio, IntelliSense

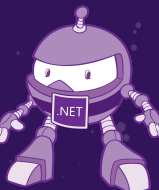


Blazor

<https://blazor.net>



A .NET SPA framework, runs on the browser via wasm.



“Blazor” experimental project

- Requires no plugin or code transpilation
- Works in all modern browsers including mobile browsers
- Browser + Razor = Blazor!



.CS



.cshtml

Compile to .NET
assemblies

BROWSER

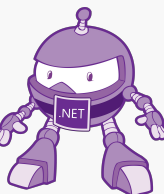
App.dll

.NET
(mscorlib.dll,
System.Core.dll,...)

WebAssembly
(mono.wasm)

Get started with Blazor

- Go to <https://blazor.net>
- Install the .NET Core SDK (<https://dot.net>)
- Install the Blazor CLI templates
 - `dotnet new -i Microsoft.AspNetCore.Blazor.Templates`
- In Visual Studio, install the Blazor extension
 - <https://marketplace.visualstudio.com/items?itemName=aspnet.blazor>



Blazor features

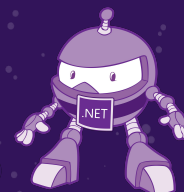
- Component model
- Dependency injection
- JavaScript interop
- Fallback to asm.js

Demo

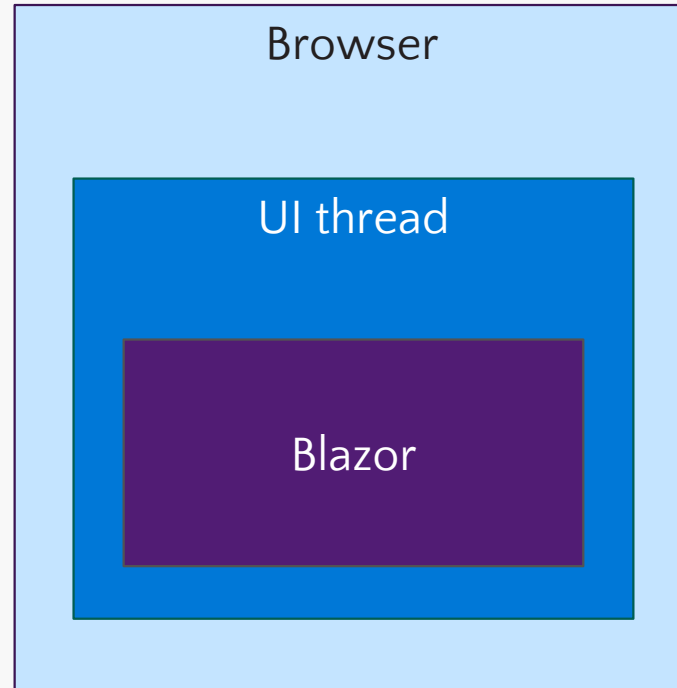
.NET



.NET Conf 2018



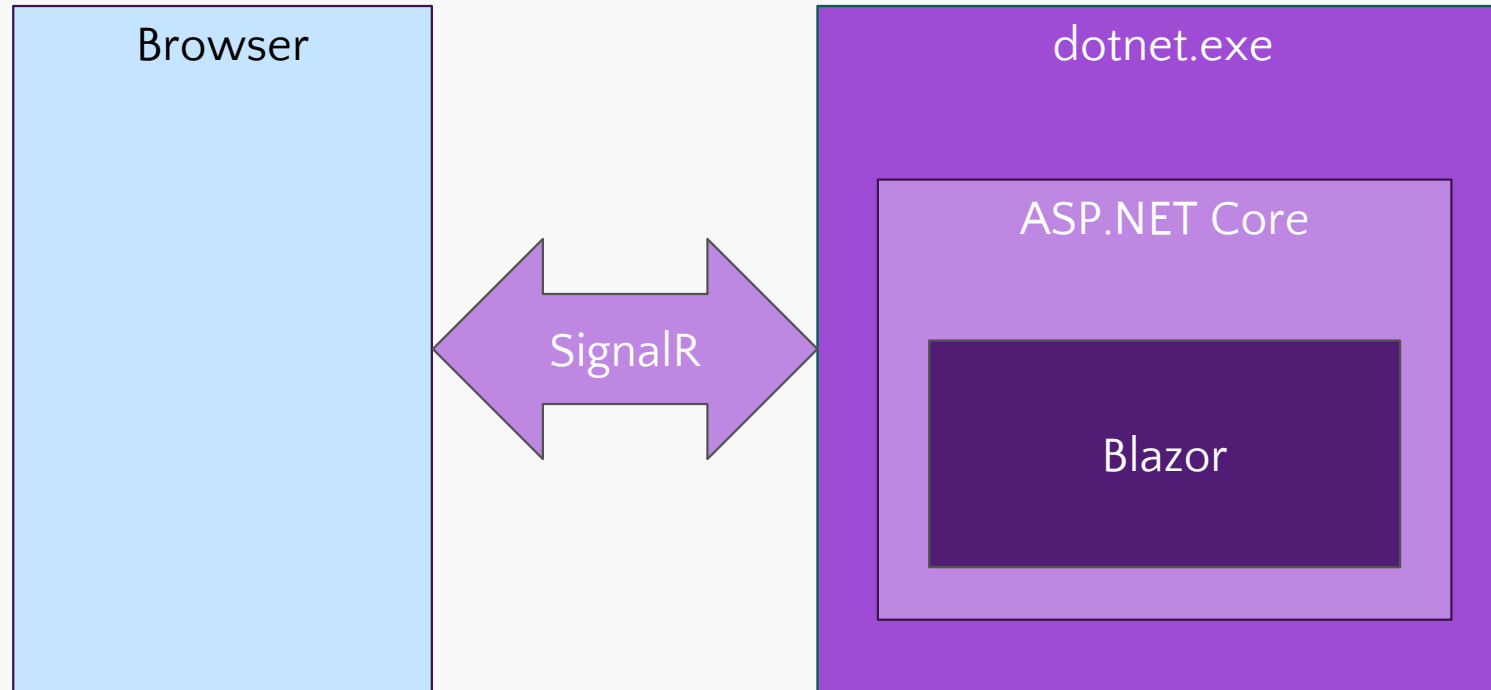
Client-side Blazor



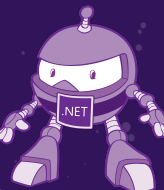
It's experimental! Lots to do

- Runtime performance
- AoT compilation to WebAssembly
- Optimize download size
- Improved .NET Standard support
- Live reload
- Server-side rendering
- Debugging

Server-side Blazor



Server-Side Demo.NET



Razor Components (“server-side Blazor”)

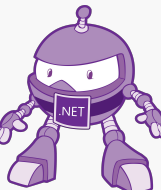
- Blazor programming model integrated into ASP.NET Core
- Runs server-side on .NET Core
- Handle client UI interactions using SignalR
- Debugging & existing libraries “just work”
- Same code can run client-side on WebAssembly with no code changes
- Shipping in .NET Core 3.0

Blazor roadmap

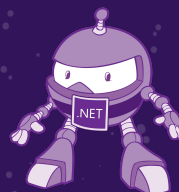
- ~~Sept – Blazor 0.6.0~~
- ~~Monthly – Ongoing Blazor updates~~
- ~Early 2019 – Preview Razor Components in .NET Core 3.0

Additional resources

- Docs: <https://blazor.net>
- Code: <https://github.com/aspnet/blazor>
 - Mono: <https://github.com/mono/mono>
- Community: <https://blazor.net/community>
 - Samples, libraries, courses, tooling, etc.
- Chat: <https://gitter.im/aspnet/blazor>



Q&A



A purple robot character with a rounded head, a small antenna on top, and a friendly expression. It is holding a small rectangular sign in front of its chest that displays the text ".NET". The robot has two arms and two legs, all in a matching purple color. The background is a dark purple with some lighter purple bokeh-like spots.