

Is ASP.NET Core Ready for Primetime?

Hi, I'm J.

Jonathan "J." Tower

Partner & Principal Consultant

Trailhead Technology Partners



Microsoft MVP in ASP.NET



Telerik/Progress Developer Expert



Organizer of Beer City Code



TRAILHEAD
TECHNOLOGY PARTNERS

trailheadtechnology.com



jtower@trailheadtechnology.com



trailheadtechnology.com/blog



[jtowermi](https://twitter.com/jtowermi)

github.com/jonathantower/aspnet-core-primetime

If You Give \$500, So Will I!

bit.ly/stirtrek-water

"charity:water is a non-profit organization that provides clean and safe drinking water to people in developing nations. The organization was founded in 2006 and has helped fund 22,936 projects in 24 countries, benefiting over 4.6 million people." - Wikipedia

*"4/4 Stars"
- CharityNavigator.org*



charity: water

Is ASP.NET Core
Ready for Primetime?

Yup!

Thank You! Questions?*

*Just kidding

Some History

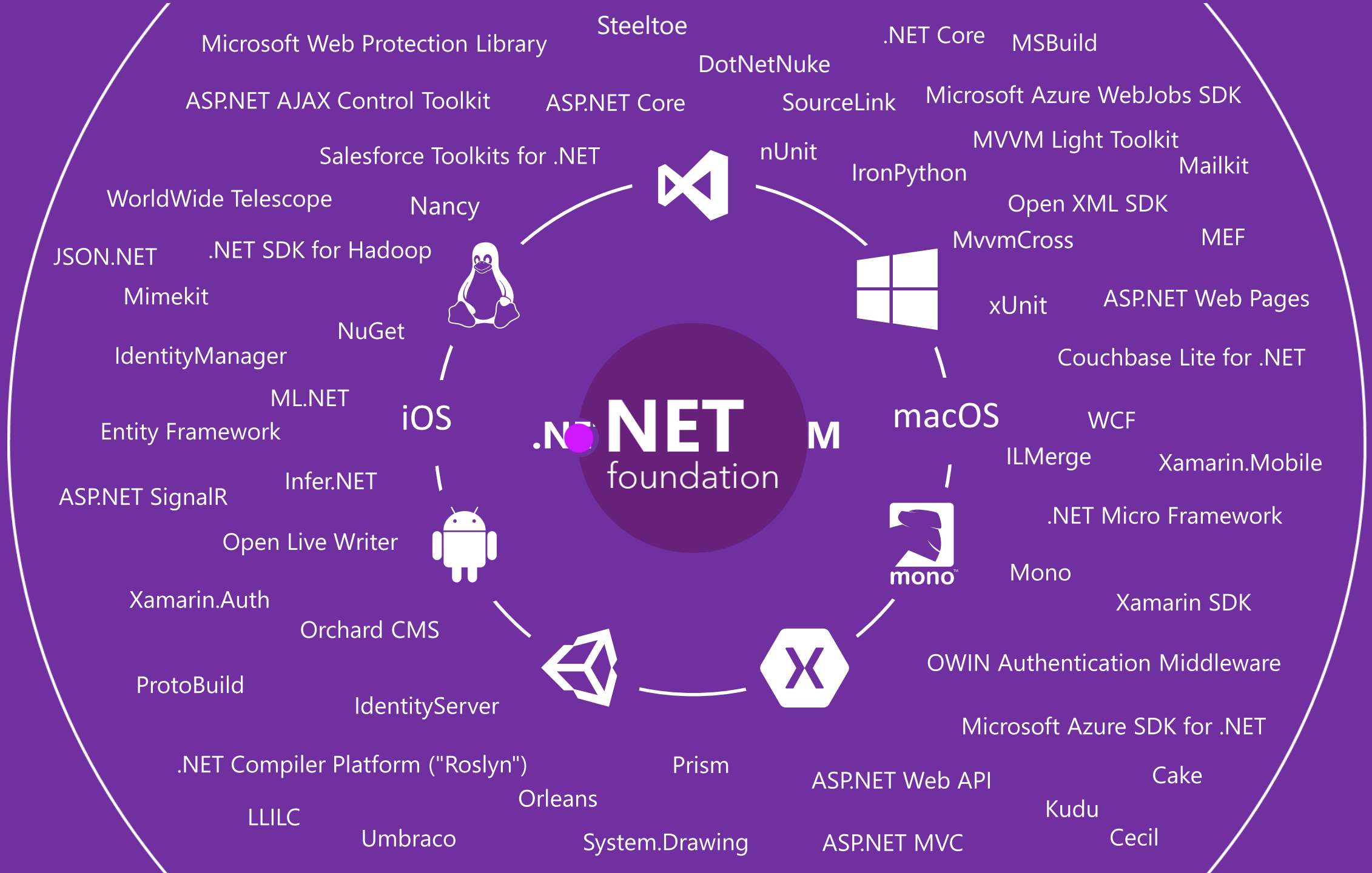
History of ASP.NET Core

- ASP.NET vNext May 2014
- ASP.NET 5 Feb 2015
- ASP.NET Core 1.0 January (released June) 2016
- ASP.NET Core 1.1 November 2017
- ASP.NET Core 2.0 August 2017
- ASP.NET Core 2.1 May 2018
- ASP.NET Core 2.2 December 2018
- ASP.NET Core 3.0 2nd half 2019 (TBA at Build)

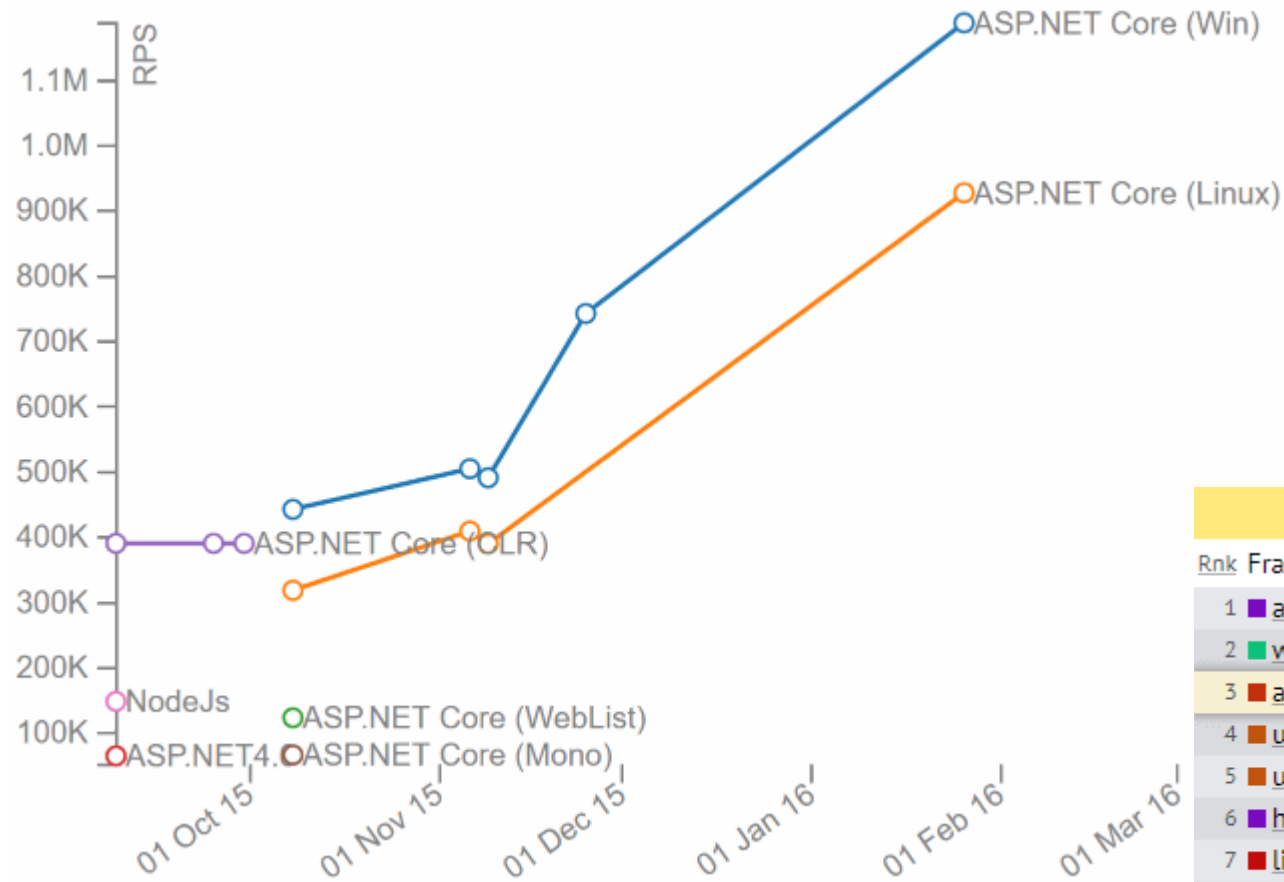
The Reasons ASP.NET Core is Ready for Prime-Time





















Open-Source



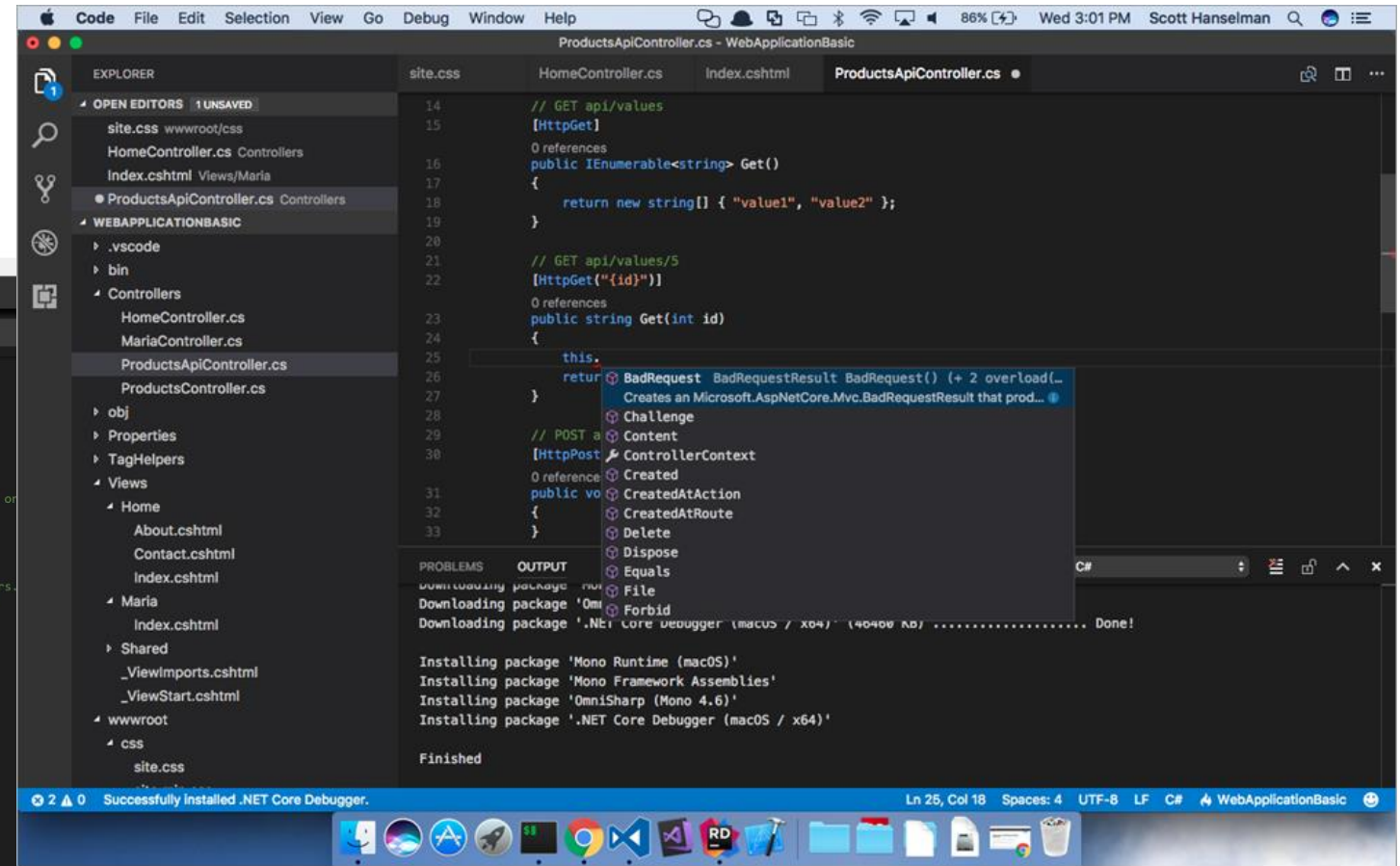
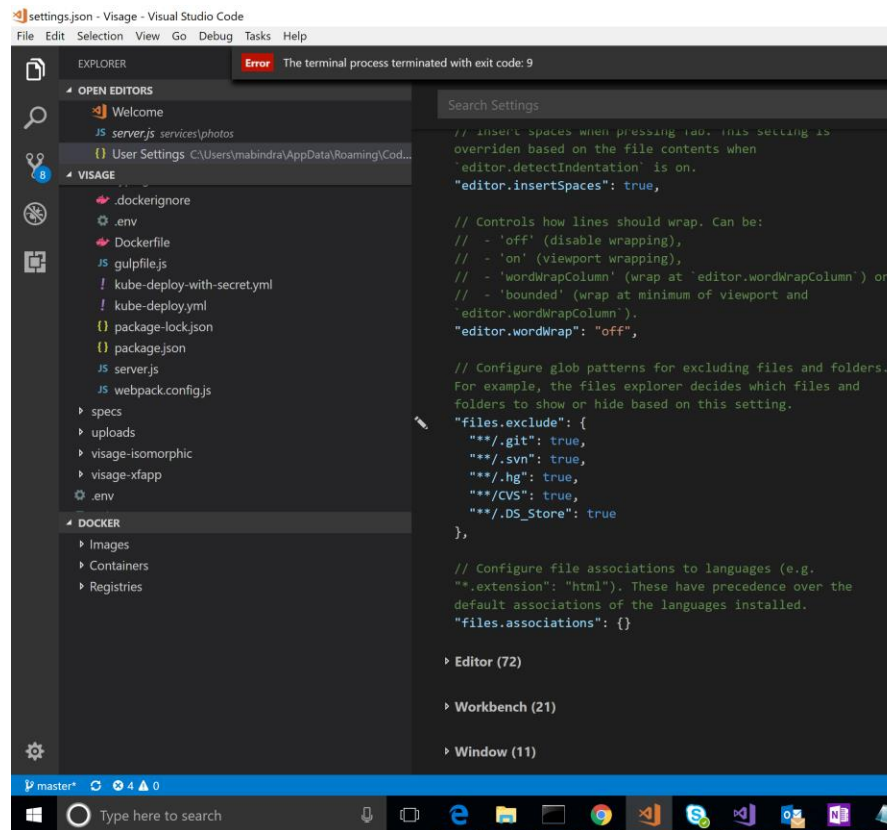


Performance



Best plaintext responses per second, Test environment (331 tests)							
Rnk	Framework	Best performance (higher is better)		Cls	Lng	Plt	FE
1	 actix-raw	7,003,320	 100.0%	Plt	Rus	Non	act
2	 wizzardo-http	7,002,116	 100.0%	Mcr	Jav	Non	Non
3	 aspcore	7,000,118	 100.0%	Plt	C#	.NE	kes
4	 ulib	6,998,365	 99.9%	Plt	C++	Non	ULi
5	 ulib-plaintext_fit	6,998,068	 99.9%	Plt	C++	Non	ULi
6	 hyper	6,996,874	 99.9%	Mcr	Rus	Rus	Hyp
7	 libreactor	6,996,726	 99.9%	Mcr	C	Non	Non
8	 tokio-minihttp	6,995,981	 99.9%	Mcr	Rus	Rus	tok
9	 baseio-http-lite	6,983,540	 99.7%	Plt	Jav	bas	Non
10	 aspcore-rhtx	6,975,733	 99.6%	Plt	C#	.NE	kes

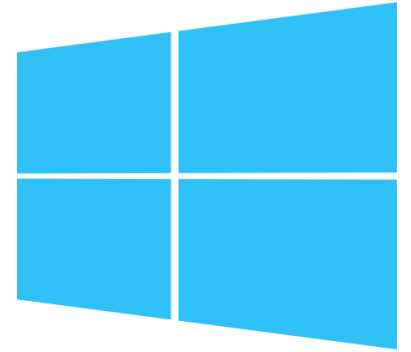
Cross-Platform Development



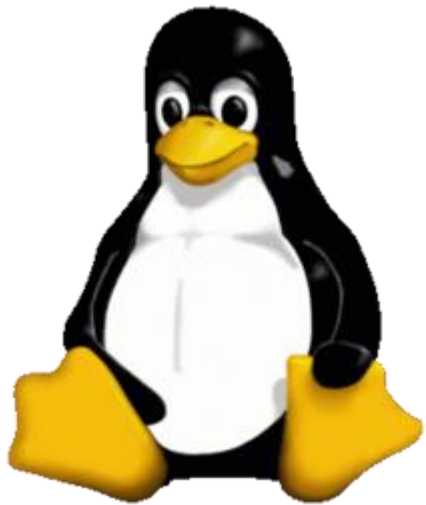
Cross-Platform Development

DEMO: Running ASP.NET Core from Linux

Cross-Platform Hosting



Apache



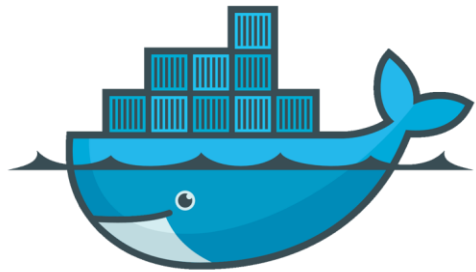
Mac OS

NGINX

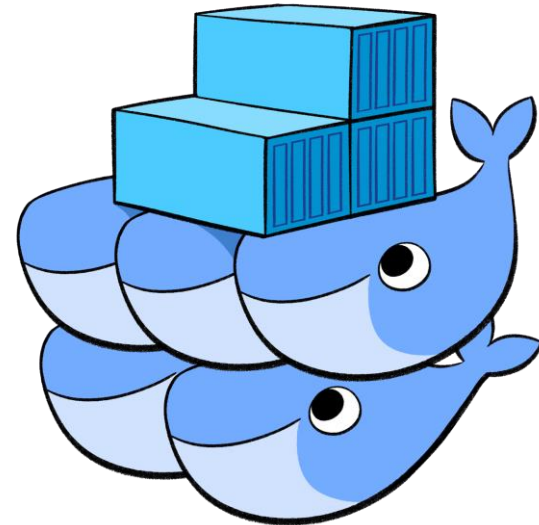
Containerization Support



kubernetes



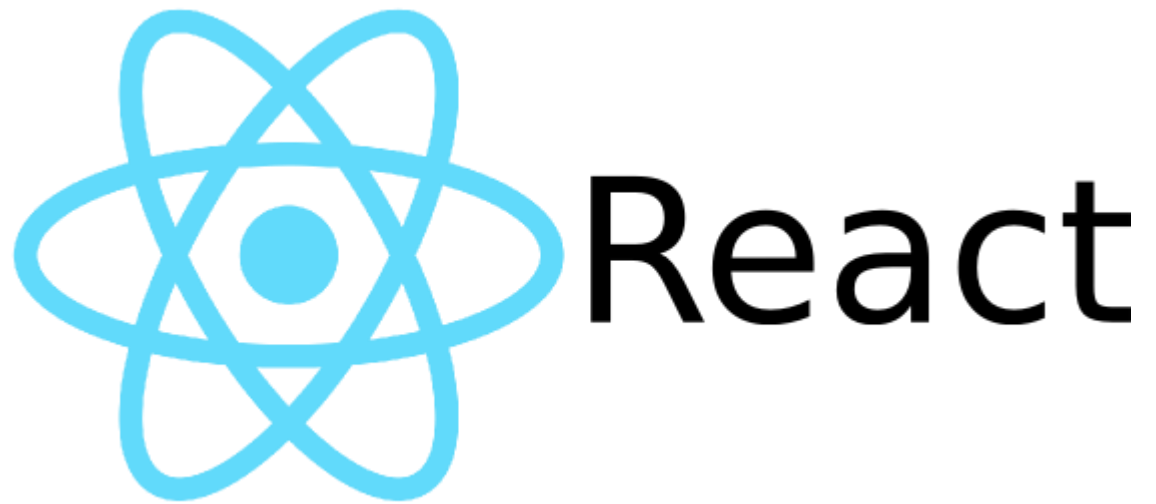
docker



Containerization Support

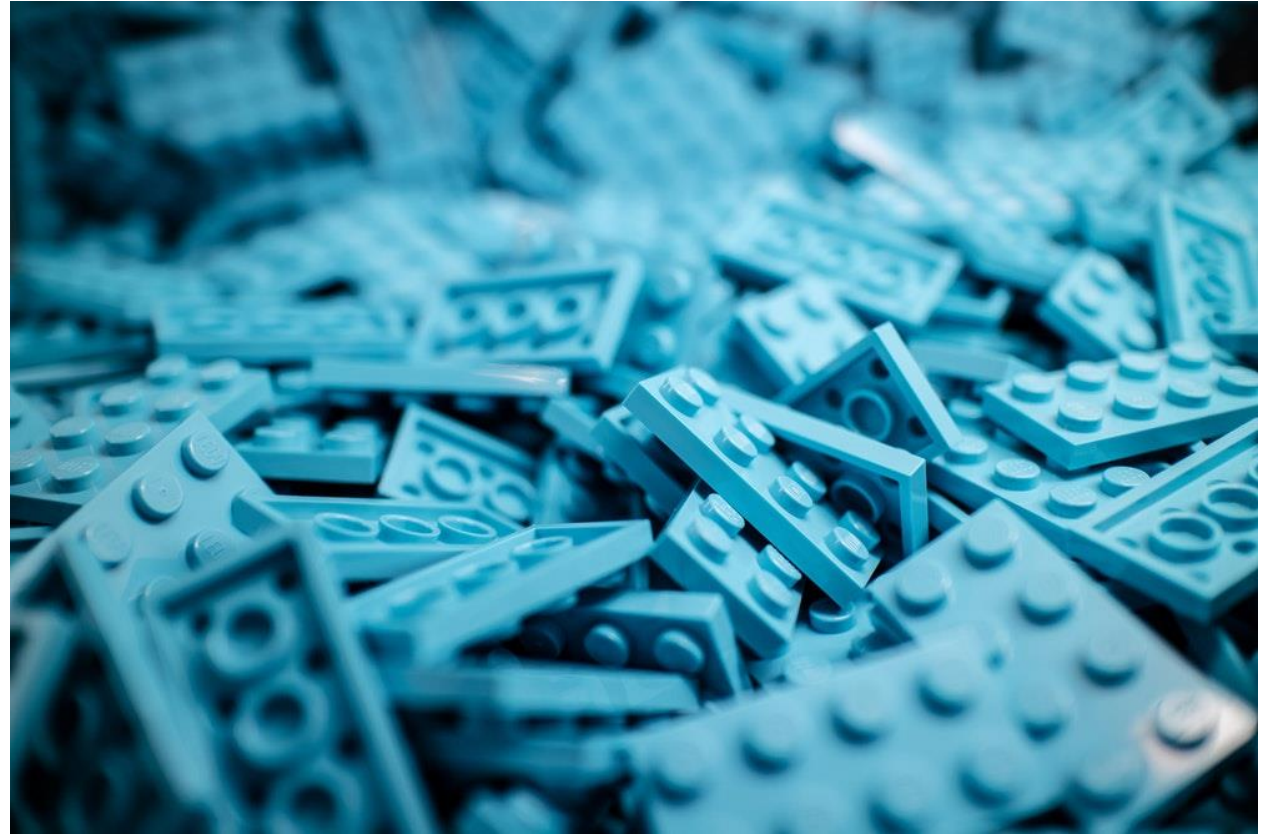
DEMO: Using Docker

JavaScript SPA Framework Support



Modularization

- Everything is in Nuget packages
- Only use what you need
- No dependency on monolithic framework
- Allows for each side-by-side deployment



.NET Standard 2.0 and Beyond

- .NET Standard is for sharing code libraries
- Replaced PCLs
- 250% more APIs than 1.6
- .NET Standard 2.0 => .NET Framework 4.7.2
- .NET Standard 2.0 => .NET Core 2.0

Unified MVC and Web API



- One namespace, one set of framework classes
- Single controller base class
- Shared authentication stack
- Same model-binding rules
- [ApiController] attribute
 - Auto model validation
 - Model binding inference
 - Other small differences

Dependency Injection

- Dependency injection built-in
- DI is already used by the framework
- Simple lifetime options:
 - Singleton
 - Request (Scoped)
 - Transient
- 3rd party DI framework integrate easily



kestrel



- Decouples ASP.NET Core from IIS
- Cross platform
- Built for speed
- Limited features (port sharing)

Tag Helpers

- Think of it as AngularJS directive for server-side programming

DEMO: Custom email link tag helper

Razor Pages

- Paged-based model built on ASP.NET Core MVC
- More MVVM than MVC
- Simpler, more easy to organize



Side-by-side Versioning



- Framework-dependent vs self-contained
- No worries about missing features
- Deploy without fear of breaking old apps

Configuration

- JSON based
- Also supports XML, Azure key valut, command line, environment variables, environment files
- Easier hierarchical support
- `IOptions<T>`

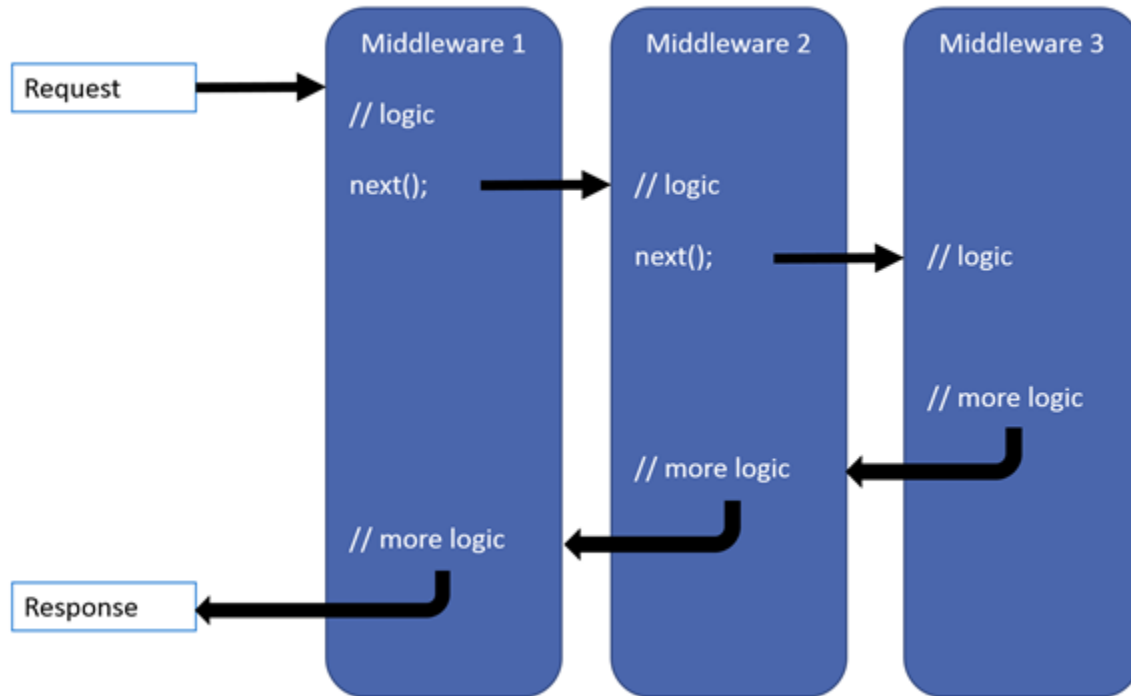


Logging



- Built-in and 3rd party providers
 - Console, Debug, EventSource, EventLog, TraceSource
 - AzureAppServices, ApplicationInsights
 - elmah, Log4net, NLog, Serilog, others...

Middleware



- A little like HTTPModules
- Configured in code (not XML)
- Easy control over the order

DEMO: Timing middleware

What's Still Missing?

EF Core

- Inheritance: Table per type (TPT)
- Inheritance: Table per concrete class (TPC)
- Many-to-many without join entity
- Create model from database: VS wizard
- Update model from database
- Text-based query language (Entity SQL)
- Change tracking: Proxies
- Lifecycle hooks (events, interception)
- Simple Logging (Database.Log)

Full HTTP/2 and HTTP/3 Support

- HTTP/2 missing on Mac
- HTTP/3 (QUIC)

Web Forms

- Likely never to be ported
- Razor Pages is most similar
- Consider a more modern web app



Other Little Things

- Built-in LetsEncrypt support
- Authorization for static files
- Response Caching with Redis or without cookies

Summing Up

Low Risk, High Reward

- Modern
- Open source
- Cross-platform
- High performance
- Stable since 2.0
- Full-featured since .NET Standard 2.0
- Side-by-side versioning

.NET Framework is D#@d



- Supported for a very long time
- Ubiquitous on Windows
- Not cross-platform
- Old, slow compared to Core
- From BUILD: all new feature development will be done in .NET Core

.NET Standard	1.0	1.1	1.2	1.3	1.4	1.5	1.6	2.0	2.1
.NET Core	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	3.0
.NET Framework	4.5	4.5	4.5.1	4.6	4.6.1	4.6.1 ¹	4.6.1 ¹	4.6.1 ¹	N/A ²
Mono	4.6	4.6	4.6	4.6	4.6	4.6	4.6	5.4	6.2
Xamarin.iOS	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.14	12.12
Xamarin.Mac	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.8	5.12
Xamarin.Android	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	9.3
Unity	2018.1	2018.1	2018.1	2018.1	2018.1	2018.1	2018.1	2018.1	TBD
Universal Windows Platform	8.0	8.0	8.1	10.0	10.0	10.0.16299	10.0.16299	10.0.16299	TBD

¹ The versions listed here represent the rules that NuGet uses to determine whether a given .NET Standard library is applicable. While NuGet considers .NET Framework 4.6.1 as supporting .NET Standard 1.5 through 2.0, there are several issues with consuming .NET Standard libraries that were built for those versions from .NET Framework 4.6.1 projects. For .NET Framework projects that need to use such libraries, we recommend that you upgrade the project to target .NET Framework 4.7.2 or higher.

² .NET Framework will not support .NET Standard 2.1 or any other later version. For more details, see this [blog post](#).

Source: <https://github.com/dotnet/standard/blob/master/docs/versions.md>

New Development on .NET Core

"So you shouldn't feel pressured to move off .NET Framework, but just know that it's going to be much more highly targeted compatible fixes kind of going forward and **we recommend that all new development that you start on .NET is on .NET Core** if possible."



Beth Massi
Product Marketing Manager
.NET Platform, Microsoft
@bethmassi

Migrating from ASP.NET to ASP.NET Core

Somewhat migratable:

- SPAs
- MVC
- Web API

Must be replaced:

- Web Forms (Razor Pages?)



If You Give \$500, So Will I!

bit.ly/stirtrek-water

“charity:water is a non-profit organization that provides clean and safe drinking water to people in developing nations. The organization was founded in 2006 and has helped fund 22,936 projects in 24 countries, benefiting over 4.6 million people.” - Wikipedia

“4/4 Stars”

- CharityNavigator.org



charity: water

Thank You! Questions?

Jonathan "J." Tower

Partner & Principal Consultant

Trailhead Technology Partners



🏆 Microsoft MVP in ASP.NET

🏆 Telerik/Progress Developer Expert

📅 Organizer of Beer City Code

✉ jtower@trailheadtechnology.com

🌐 trailheadtechnology.com/blog

🐦 jtowermi

github.com/jonathantower/aspnet-core-primetime