

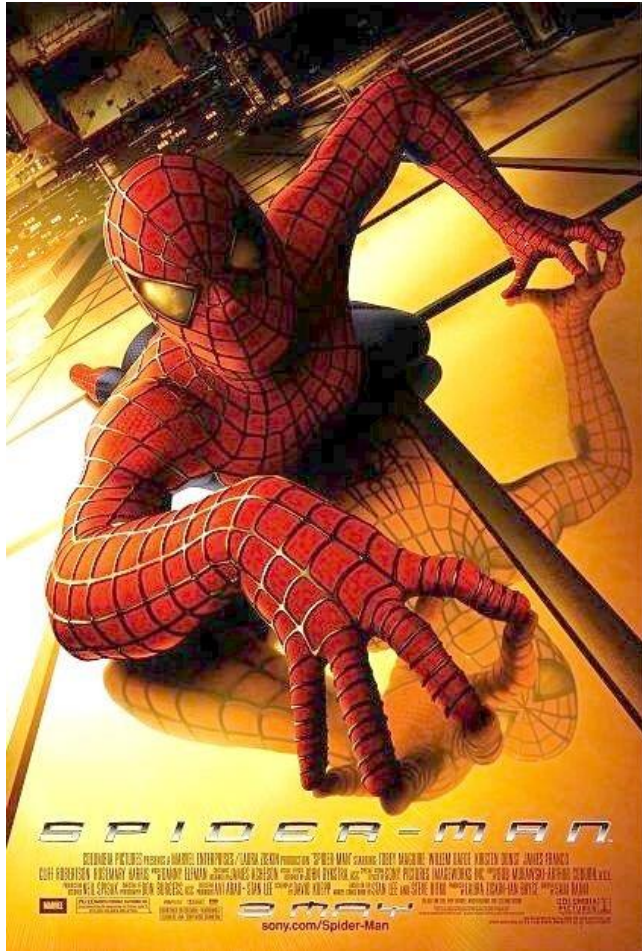
What's new in ASP.NET Core 1.0

(Formerly known as ASP.NET 5)

Israel Dot NET Developers User Group

April 2016

Flashback from 2002



Microsoft®
.net™



So what has changed?



Where's Waldo?

Yonatan Mevorach

.NET Web Developer

at
sears
.israel



.NET Core Overview

- Open-sourced (MIT)
- Cross-platform
- Portable
- Modular
- "CLI first"
- Performance

- A "Reboot"

Vision

- “The audience is the 15-year-old girl who went to a hackathon... so then fast forward 10 or 15 years, when that 15-year-old girl starts facebook v2- .net's in the running”
 - Scott Hanselman
- “We need to get those things right to enable the next 5, 10, 15 years of innovation on .net”
 - Damian Edwards

Open-sourced (MIT)

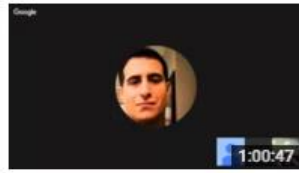
- <https://github.com/dotnet/core> (Samples)
- <https://github.com/dotnet/coreclr> (Runtime)
- <https://github.com/dotnet/corefx> (Libraries)
- <https://github.com/dotnet/cli> (Command Line)

- [.NET Core Design Reviews Videos](#)

Open-sourced



.NET Core Design Reviews: TypeDescriptors
67 views • 1 month ago



.NET Core Design Reviews: BufferPool in CoreFX and...
49 views • 1 month ago



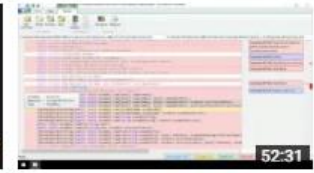
.NET Core Design Reviews: ADO.NET
114 views • 2 months ago



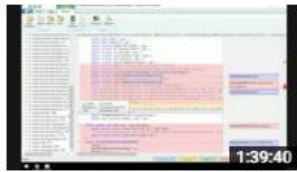
.NET Core Design Reviews: Roslyn's IOperation
154 views • 2 months ago



.NET Core Design Reviews: ADO.NET
120 views • 2 months ago



.NET Core Design Reviews: Bringing back Reflection APIs (3)
39 views • 2 months ago



.NET Core Design Reviews: Bringing back Reflection APIs (2)
72 views • 2 months ago



.NET Core Design Reviews: Bringing back Reflection APIs
71 views • 2 months ago



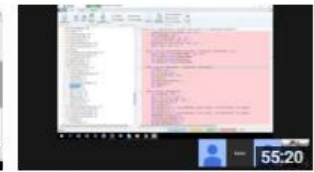
.NET Core Design Reviews: Finalization during shutdown
44 views • 3 months ago



.NET Core Design Reviews: SIMD and Console
108 views • 3 months ago



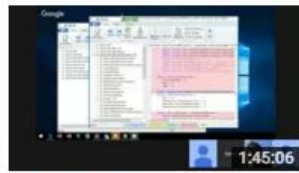
.NET Core Design Reviews: Backlog review
53 views • 3 months ago



.NET Core Design Reviews: Misc
204 views • 4 months ago



.NET Core Design Reviews: Backlog review
31 views • 4 months ago



.NET Core Design Reviews: Networking APIs
118 views • 4 months ago



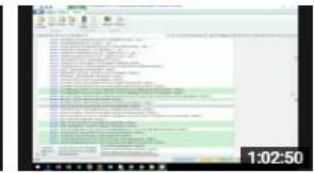
.NET Core Design Reviews: ManagedBufferPool
66 views • 4 months ago



.NET Core Design Reviews: Misc
41 views • 5 months ago



.NET Core Design Reviews: C# Interactive APIs
133 views • 5 months ago



.NET Core Design Reviews: Portable PDB
89 views • 6 months ago



.NET Core Design Reviews: Backlog & Serialization
112 views • 6 months ago



.NET Core Design Reviews: ITelemetry*, INotify, ...
125 views • 6 months ago



.NET Core Design Reviews: Backlog Review
57 views • 7 months ago



.NET Core Design Reviews: MAX_PATH Review
71 views • 7 months ago



.NET Core Design Reviews: MAX_PATH Review
78 views • 7 months ago



.NET Core Design Reviews: TryGetRawMetatadata
69 views • 7 months ago

Cross-platform

It is very easy to get started with .NET Core on your platform of choice.

You just need a shell, a text editor and 10 minutes of your time. Ready? Set? Let's go!

Windows	Ubuntu (14.04)	OS X (10.11)	Docker
---------	----------------	--------------	--------

	Debug	Release
CentOS 7.1	x64 passing	x64 building
Debian 8.2	x64 passing	x64 passing
FreeBSD 10.1	x64 building	x64 building
openSUSE 13.2	x64 passing	x64 passing
OS X 10.11	x64 building	x64 building
Red Hat 7.2	x64 passing	x64 passing
Ubuntu 14.04	x64 building	x64 building
Ubuntu 15.10	x64 building	x64 passing
Windows 8.1	x64 building	x64 building
	arm64 passing	arm64 passing

Modular & Portable

- Application-local model instead of a Machine-wide model
- Enables releasing updates in a much more agile fashion
- [Blogpost: Introducing .NET Core](#)

Bye Bye GAC (Global Assembly Cache)



What's missing

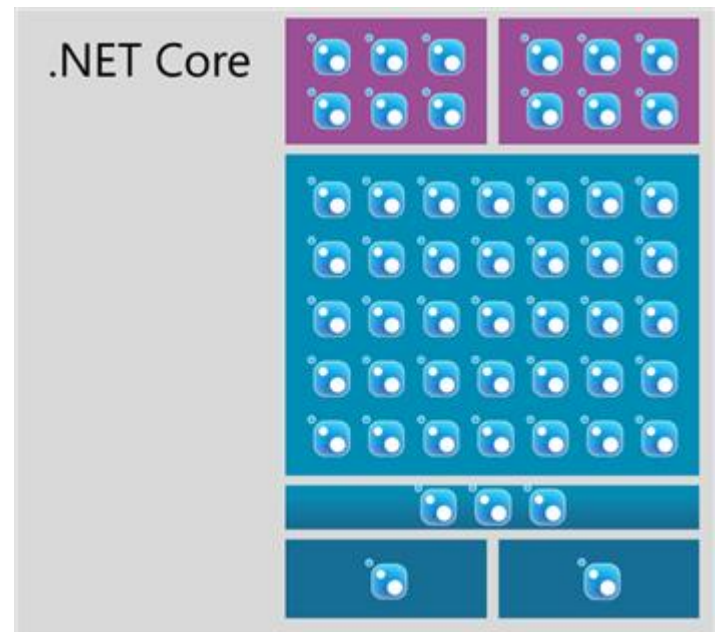
- System.Data (partly)
- System.Drawing
- System.Xml.Xsl
- System.Xml.Schema
- System.Net.Mail
- System.IO.Ports
- ... and more

- [Blogpost: Porting to .NET Core](#)

NuGet



- The primary experience for references and dependencies
- No difference between 1st party .net dependencies and 3rd party dependencies
- .NET Core is made up of many NuGet packages



NuGet



- Use the right tool for the right job
 - NuGet is for .net code
 - NPM \ Bower is for scripts (jQuery, Angular, etc.)
- GUI-less, autocomplete, auto download mode

Updating dependencies



NuGet: WebApplication10

Browse Installed Updates

Search (Ctrl+E) Include prerelease

Package source:

	Antlr by Terence Parr	v3.4.1.9004
	ANother Tool for Language Recognition, is a language tool that provides a framework for constructing recognizers, interpreters, compilers, and translators from grammatical descriptions...	v3.5.0.2
	bootstrap by Mark Otto, Jacob Thornton	v3.0.0
	Sleek, intuitive, and powerful front-end framework for faster and easier web development.	v3.3.6
	EntityFramework by Microsoft	v6.1.3
	Entity Framework is Microsoft's recommended data access technology for new applications.	
	jQuery by jQuery Foundation, Inc.	v1.10.2
	jQuery is a new kind of JavaScript Library. jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event...	v2.2.1
	jQuery.Validation by Jörn Zaefferer	v1.11.1 <input checked="" type="radio"/> <input checked="" type="button" value="✗"/>
	This jQuery plugin makes simple clientside form validation trivial, while offering lots of option for customization. That makes a good choice if you're building something new from scratch, but also...	v1.14.0 <input type="button" value="↑"/>
	Microsoft.AspNet.Identity.Core by Microsoft	v2.2.1
	Core interfaces for ASP.NET Identity.	
	Microsoft.AspNet.Identity.EntityFramework by Microsoft	v2.2.1
	ASP.NET Identity providers that use Entity Framework.	

Updating dependencies



NuGet: WebApplication10

Browse Installed **Updates 10**

Search (Ctrl+E) Include prerelease

Package source:

Select all packages

<input checked="" type="checkbox"/>	Antlr by Sam Harwell, Terence Parr ANother Tool for Language Recognition, is a language tool that provides a framework for constructing recognizers, interpreters, compilers, and translators from grammatical descrip...	<input checked="" type="checkbox"/> v3.4.1.9004 v3.5.0.2
<input checked="" type="checkbox"/>	bootstrap by Twitter, Inc. Bootstrap framework in CSS. Includes fonts and JavaScript	<input checked="" type="checkbox"/> v3.0.0 v3.3.6
<input checked="" type="checkbox"/>	jQuery by jQuery Foundation, Inc. jQuery is a new kind of JavaScript Library. jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, e...	<input checked="" type="checkbox"/> v1.10.2 v2.2.1
<input checked="" type="checkbox"/>	jQuery.Validation by Jörn Zaefferer This jQuery plugin makes simple clientside form validation trivial, while offering lots of option for customization. That makes a good choice if you're building something new from scratch, b...	<input checked="" type="checkbox"/> v1.11.1 v1.14.0
<input checked="" type="checkbox"/>	Microsoft.CodeDom.Providers.DotNetCompilerPlatform by Microsoft Replacement CodeDOM providers that use the new .NET Compiler Platform ("Roslyn") compiler as a service APIs.	<input checked="" type="checkbox"/> v1.0.0 v1.0.1
<input checked="" type="checkbox"/>	Microsoft.Net.Compilers by Microsoft .Net Compilers package. Referencing this package will cause the project to be built using the specific version of the C# and Visual Basic compilers contained in the package, as opposed to an...	<input checked="" type="checkbox"/> v1.0.0 v1.1.1
<input type="checkbox"/>	Modernizr by Faruk Ates, Paul Irish, Alex Sexton	<input checked="" type="checkbox"/> v2.6.2

Future of dependencies in VS

@VisualStudio

Bye Bye .csproj



MERGE
CONFLICTS

project.json

Replaces .csproj file

- JSON instead of XML
- All files in the dir are part of the project by default
- Supports globbing

```
"exclude": [  
  "wwwroot",  
  "node_modules"  
],
```

project.json

- Dependencies (NuGet, other projects)

```
"dependencies": {  
  "EntityFramework.Commands": "7.0.0-rc1-final",  
  "ClassLibrary1": "1.0.0-*"  
},
```

- Frameworks

```
"frameworks": {  
  "netcoreapp1.0": { }  
}
```

- [Docs: Project.json file](#)

CLI First

- Use whatever IDE \ Editor you want using OmniSharp



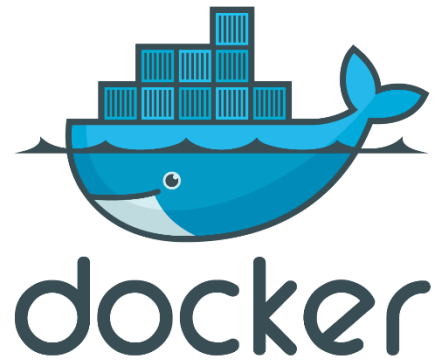
- Build and run with the dotnet CLI tool

```
Common Commands:
new           Initialize a basic .NET project
restore      Restore dependencies specified in the .NET project
build        Builds a .NET project
publish      Publishes a .NET project for deployment (including the runtime)
run          Compiles and immediately executes a .NET project
test         Executes tests in a test project
repl        Launch an interactive session (read, eval, print, loop)
pack         Creates a NuGet package
```

- [Getting started with .NET Core](#)

DEMO

Hello World



Offtopic: .NET Native

- A tool to compile IL bytecode into native machine code Ahead of Time (AOT)
- Great performance implications
- In the early stages of development
- [About: .NET Ecosystem](#)
- [Github: .NET Core Runtime \(CoreRT\)](#)

ASP.NET Core 1.0

- Open-sourced (Apache)
- Cross-platform
- Portable
- Modular
- "CLI first"
- Performance

Open-sourced

- <https://github.com/aspnet/Home>
- <https://github.com/aspnet/Mvc>
- <https://github.com/aspnet/DependencyInjection>
- <https://github.com/aspnet/Razor>
- <https://github.com/aspnet/EntityFramework>
- <https://github.com/aspnet/Identity>
- <https://github.com/aspnet/diagnostics/>
- <https://github.com/aspnet/configuration>
- <https://github.com/aspnet/Hosting>
- <https://github.com/aspnet/StaticFiles>
- <https://github.com/aspnet/ResponseCaching>
- <https://github.com/aspnet/KestrelHttpServer>

- [ASP.NET Community Standup](#)

Open-sourced



David Fowler

@davidfowl

 Follow

What should default the static files folder name to in **ASP.NET** Core? [#aspnetcore](#)
[@DamianEdwards](#)

34% public

57% wwwroot (don't change it)





9% What is wwwroot?

1,733 votes • Final results

ASP.NET Core 1.0

- Compatible with both .NET Core and .NET Framework 4.6
- Not tied to IIS
- MVC and Web API Unification
- Middleware
- New Configuration model
- Built-in Dependency Injection
- Tag Helpers

Run ASP.NET Core, Run

ASP.NET 4.6	ASP.NET Core 1.0
.NET Framework 4.6 	.NET Core 1.0   
.NET framework libraries	.NET core libraries
Compilers and runtime components (.NET Compiler Platform: Roslyn, C#, VB, F# Languages, RyuJIT, SIMD)	

What's missing

- SignalR
- WebForms
- Visual Basic support
- F# support

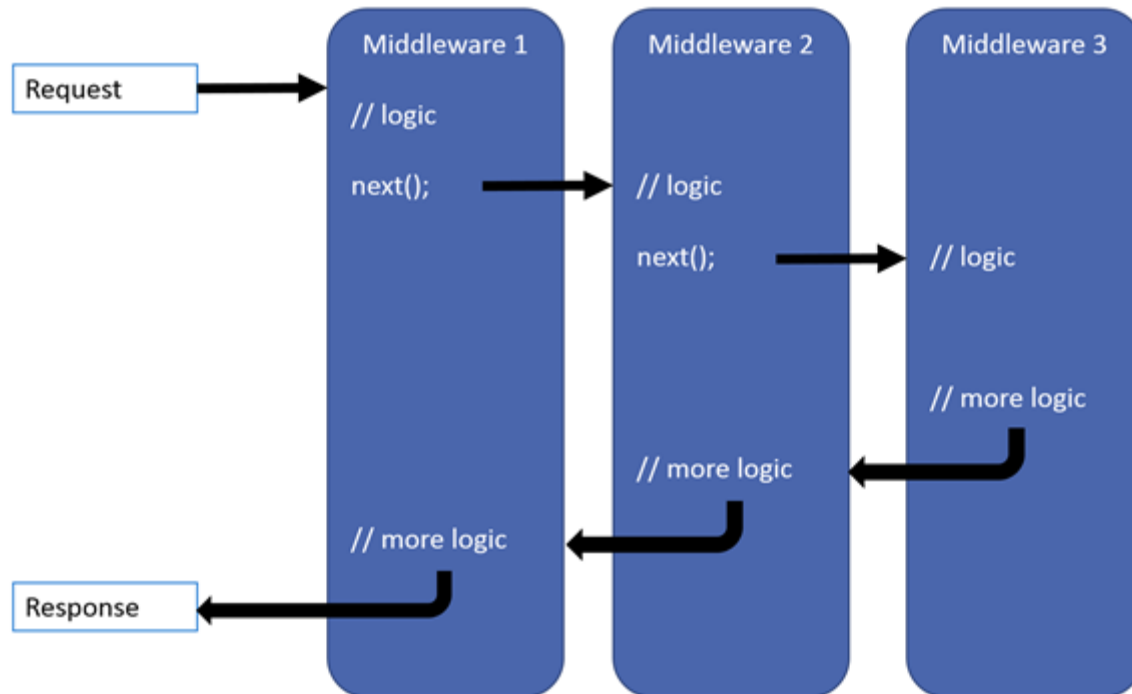
DEMO

Empty Web Application

Middleware

- A Request\Response handling component
- When put together they make up your HTTP request handling pipeline
- The order in which you set up your middleware matters! It matches the order your Request delegate will be called

Middleware



- [Docs: Middleware](#)

Middleware Everywhere!

- Almost all of the ASP.NET functionality is now implemented as Middleware (e.g. Exception Handling, Static Files, Authentication, MVC, etc.)
- Which means that
 - You can choose whether or not to use
 - Modify (since it's modular & open source)

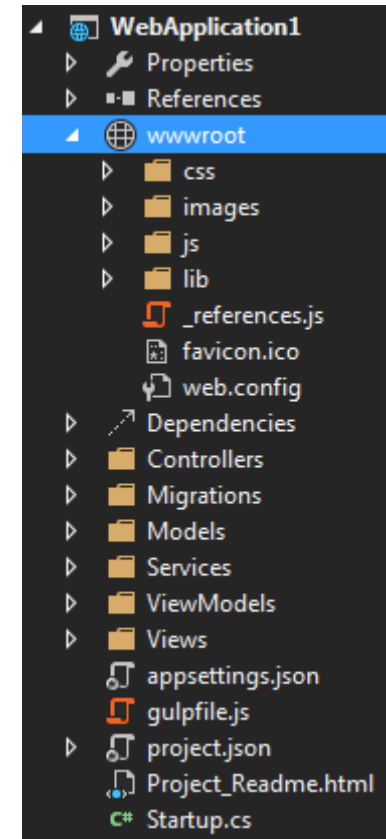
Bye Bye HTTP Handlers & HTTP Modules



- [Docs: Migrating HTTP Modules to Middleware](#)

wwwroot

- The root of the web app is now different than the root of the project
- By default, the web app root folder is called `wwwroot` (configurable), and it's under the project root folder
- Helps you protect sensitive files



DEMO

Web Application

Startup class

- When your app start it will look for a Startup class to instantiate
- And it will look for the Configure and ConfigureServices methods to call
- Startup constructor – set up Configuration
- Configure – set up Middleware
- ConfigureServices – set up Service injection
- [Source Code: StartupLoader](#)

Environment Variables

Application

Build

Debug

Configuration: Platform:

Profile:

Launch:

Command:

Application Arguments:

Working Directory:

Launch URL:

Use Specific Runtime:

Version	Platform	Architecture
<input type="text" value="1.0.0-rc1-update1"/>	<input type="text" value=".NET Framework"/>	<input type="text" value="x86"/>

Environment Variables:

Name	Value
Hosting:Environment	Development

Configuration

- The new configuration model provides is based on access to key/value pair that can be retrieved from a variety of providers
 - In memory
 - JSON file
 - Environment variables
 - XML file
 - Command line parameters
 - Custom (inherit from `ConfigurationProvider`)
- [Docs: Configuration](#)

Bye Bye web.config



Configuration

```
public Startup(IHostingEnvironment env)
{
    // Set up configuration sources.
    var builder = new ConfigurationBuilder()
        .AddJsonFile("appsettings.json")
        .AddJsonFile($"appsettings.{env.EnvironmentName}.json")
        .AddEnvironmentVariables()
        .AddUserSecrets();

    Configuration = builder.Build();
}

public IConfigurationRoot Configuration { get; set; }
```

The order of providers matters

Secret Manager

- Helps store sensitive data outside of your project tree
- Intended for use in Development mode only
 - Does not encrypt the stored secrets
- On Windows it's stored at
`%APPDATA%\microsoft\UserSecrets\\secrets.json`
- [Docs: Safe Storage of Application Secrets](#)

Nested Configuration

```
Configuration["Logging:LogLevel:System"]
```

```
Configuration.GetSection("Logging")
```

```
{  
  "ApplicationInsights": {  
    "InstrumentationKey": ""  
  },  
  "Logging": {  
    "IncludeScopes": false,  
    "LogLevel": {  
      "Default": "Verbose",  
      "System": "Information",  
      "Microsoft": "Information"  
    }  
  }  
}
```

Dependency Injection

- Dependency Injection is a technique for achieving loose coupling between classes
- ASP.NET 5 is designed from the ground up to support and leverage dependency injection
- [Blogpost: Dependency Injection in ASP.NET Core](#)

Dependency Injection

- Supports Constructor Injection
- Available service lifetime configuration
 - Transient
 - Scoped
 - Singleton
 - Instance
- [Docs: Dependency Injection](#)

Dependency Injection

- The built-in container provides a minimal feature set and is not intended to replace other containers
- Does not support
 - Registering by convention
 - Multiple implementations per interface
 - Interceptors



Dependency Injection

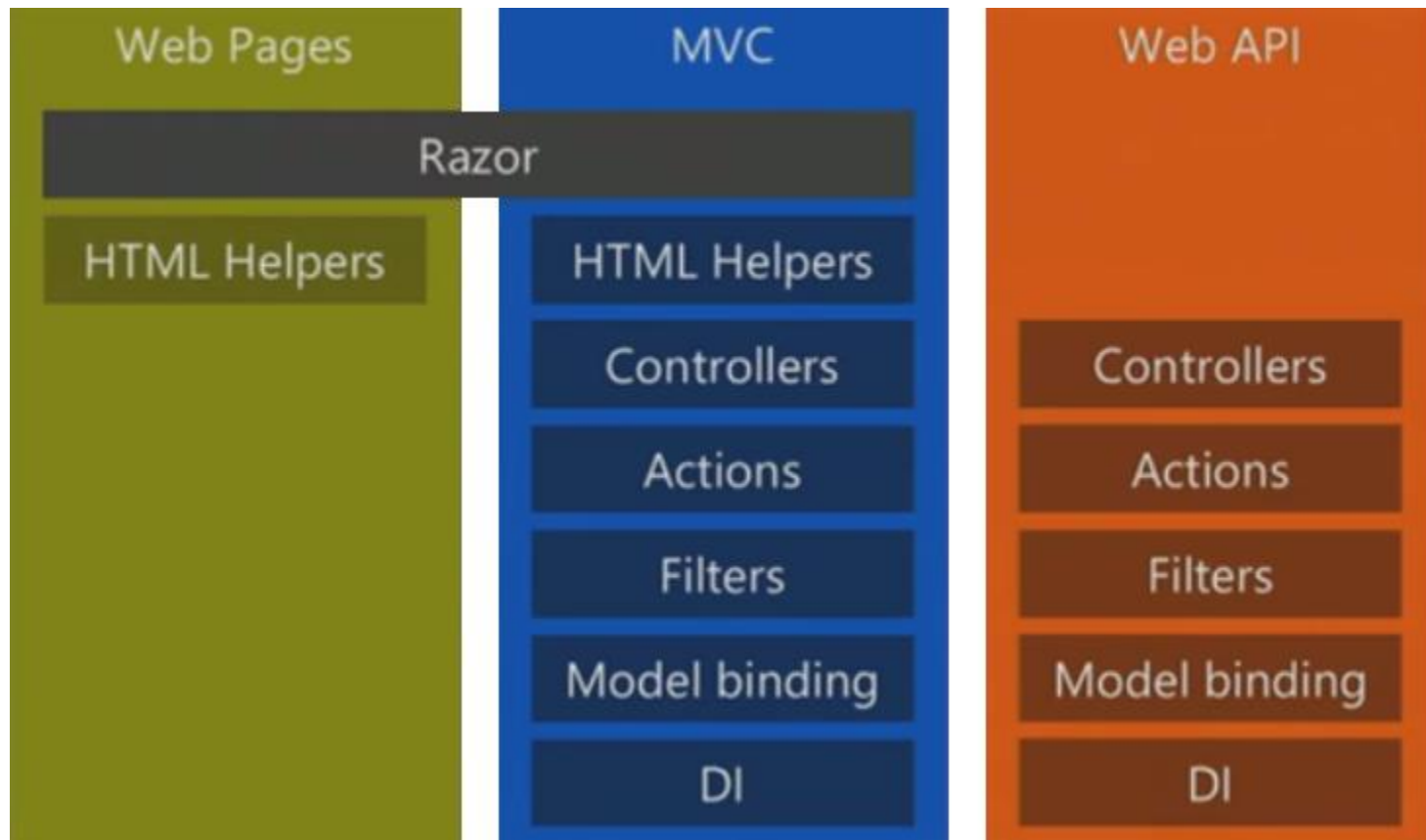
- You can replace the built-in container with any container that also implements `IServiceProvider`
- Replace `void ConfigureServices()` with `IServiceProvider ConfigureServices()`
- [Autofac](#) supports ASP.NET Core



Dynamic Development

- Dynamic compilation saves you time by removing the need to compile the app every time you change it
- Just edit, save, and refresh the browser.
- Use without Visual Studio using [dotnet-watch](#)
- [Blogpost: Introducing ASP.NET 5](#)

ASP- The story so far



MVC 6

- Unified stack for MVC and Web API apps
- It's just middleware
 - Routing
 - Razor
 - Serialization
- [Source Code: AddMvc\(\)](#)

Routing

- MVC uses a generic Routing middleware that can also be used without MVC
- Allows using inline constraints in MapRoute

```
routes.MapRoute(  
    name: "default",  
    template: "{controller=Home}/{action=Index}/{id:int?}");
```

(int, bool, datetime, float, guid, length, range, alpha, regex, required)

- Allows falling back to another route if a route handler decided it's not relevant
- [Docs: Routing](#)

Injecting services into Views

- You can also inject services into Views using the new `@inject` Razor directive

```
@inject SignInManager<ApplicationUser> SignInManager
```

- [Docs: Injecting Services Into Views](#)

View Components

- Invoked from a View and returns a Partial View
- Includes the same separation-of-concerns and testability benefits found between a controller and view

```
@Component.Invoke("FeaturedMovies")
```

- [Docs: View Components](#)
- [Sample: TagHelperSample App](#)

@await directive

- Now that we can inject Services and ViewComponents into our View, we can potentially make the rendering much slower
- The new @await Razor directive lets you render a Partial in an asynchronous manner

```
@await Html.PartialAsync("_LoginPartial")
```

FlushAsync

- Until today the Razor View Engine did not support flushing the HTTP response incrementally (Chunked Encoding)
- Now this is possible by calling `@await FlushAsync()`
- Great for perceived performance
- [Wikipedia: Chunked transfer encoding](#)

Tag Helpers

```
<div class="row">
  <div class="col-md-8">
    <section>
      <form asp-controller="Account" asp-action="Login" asp-ro
      <h4>Use a local account to log in.</h4>
      <hr />
      <div asp-validation-summary="ValidationSummary.All"
      <div class="form-group">
        <label asp-for="Email" class="col-md-2 control-l
        <div class="col-md-10">
          <input asp-for="Email" class="form-control"
          <span asp-validation-for="Email" class="text
        </div>
      </div>
      <div class="form-group">
        <label asp-for="Password" class="col-md-2 contro
        <div class="col-md-10">
```


Tag Helpers

- Everything looks like HTML
 - Plays nicely with syntax highlighting everywhere
 - But Tag Helpers aware tools (like VS) understand what's not just HTML and provide extra intellisense
- Attribute Tag Helpers render only some of the HTML attributes, not the entire element
 - You can use HTML to write HTML
- [Docs: Introduction to Tag Helpers](#)

Tag Helpers

- Attribute Tag Helpers

- `@Html.ActionLink` → `<a asp-section>`
- `@Html.BeginForm` → `<form asp-controller>`
- `@Html.LabelFor` → `<label asp-for>`

- Element Tag Helpers

- `<environment>`
Render the HTML within only in a certain env mode
- `<cache>`
Cache the rendered HTML within

Tag Helpers

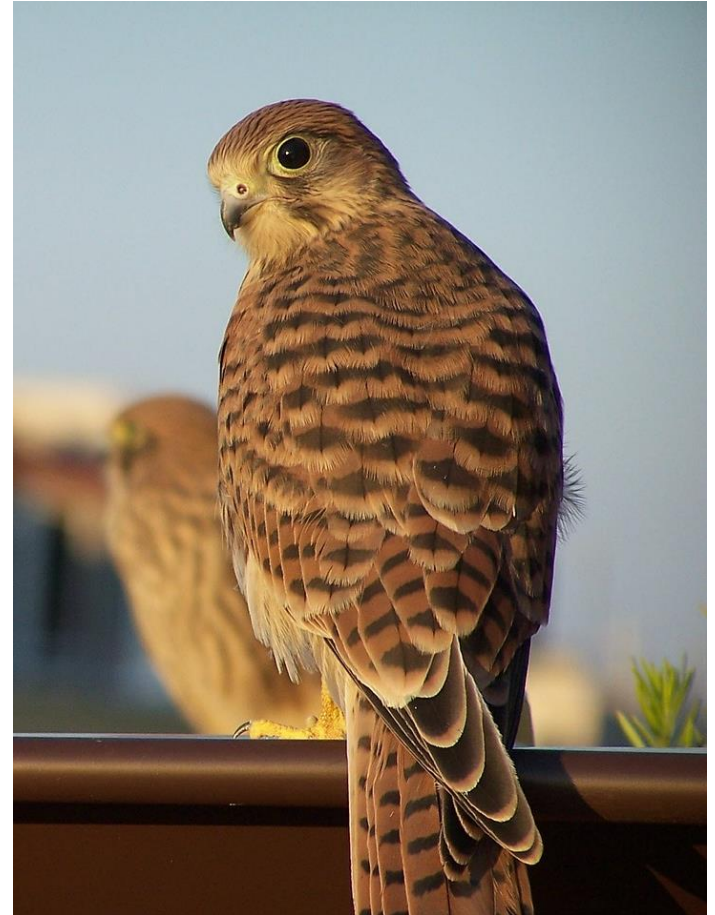
- Write your own Tag Helpers!
- Just implement the `TagHelper` class
- [Docs: Authoring Tag Helpers](#)
- [Sample: TagHelperSamples](#)

Swagger

- Swagger is a spec that defines a standard way of documenting your public API
- Based on this Swagger tools can auto-generate documentation and language-specific SDKs
- Use the [Swagger Nuget packages](#) to generate an interactive documentation site for your Web API in seconds

Kestrel

- A new web server
- Open-sourced
- Cross-platform
- Better performance
- Based on libuv
 - I/O library (network, file system)
 - Primarily developed for node.js



Kestrel

```
"commands": {  
  "web": "Microsoft.AspNet.Server.Kestrel"  
},
```

▶ web ▾

```
Hosting environment: Development  
Now listening on: http://localhost:5000  
Application started. Press Ctrl+C to shut down.
```

Kestrel

- Can run independently in production
- But it's advised to run behind a different web server in a reverse-proxy model
 - Windows: IIS
 - Using the `HttpPlatformHandler` module
 - Benefit from other IIS features (e.g. Windows Auth)
 - Linux: NGINX
- Adding new features isn't restricted by the host operating system (e.g. Websockets, HTTP 2.0)

2,300% More Requests/second
compared to ASP.NET 4.6



“ASP.NET CORE
1.15 million RPS”

<https://twitter.com/scottgu/status/700549872726839296>

KRE → DNX → dotnet CLI

- May 12, '14 - [Introducing ASP.NET vNext](#) (alpha)
- Nov 12, '14 - beta1 was [released](#)
- Nov 18, '15 - ASP.NET 5 RC1 (Release Candidate 1) was [released](#)
- Jan 19, '16 - ASP.NET 5 is [renamed](#) to ASP.NET Core 1.0
- Apr 14, '16 - Hanselman [announces](#) that RC2 is delayed in order to "re-plate" on top of the new .NET Core CLI

When to jump in?

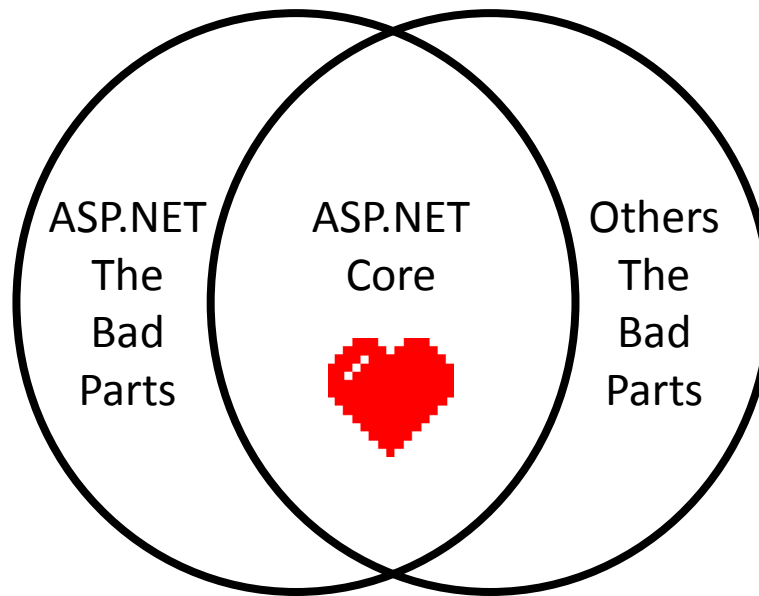
Right now* at asp.net/vnext

* But know that you're likely going to encounter out-of-date documentation (docs.asp.net), and work-in-progress code

Playing

- <https://github.com/aspnet/Home/tree/dev/samples>
- <https://github.com/aspnet/cli-samples>
- <https://github.com/Microsoft-Build-2016/CodeLabs-WebDev>
- <https://github.com/aspnet/MusicStore>

IMHO



Thank You!

@cowchimp