



Ground Up Series

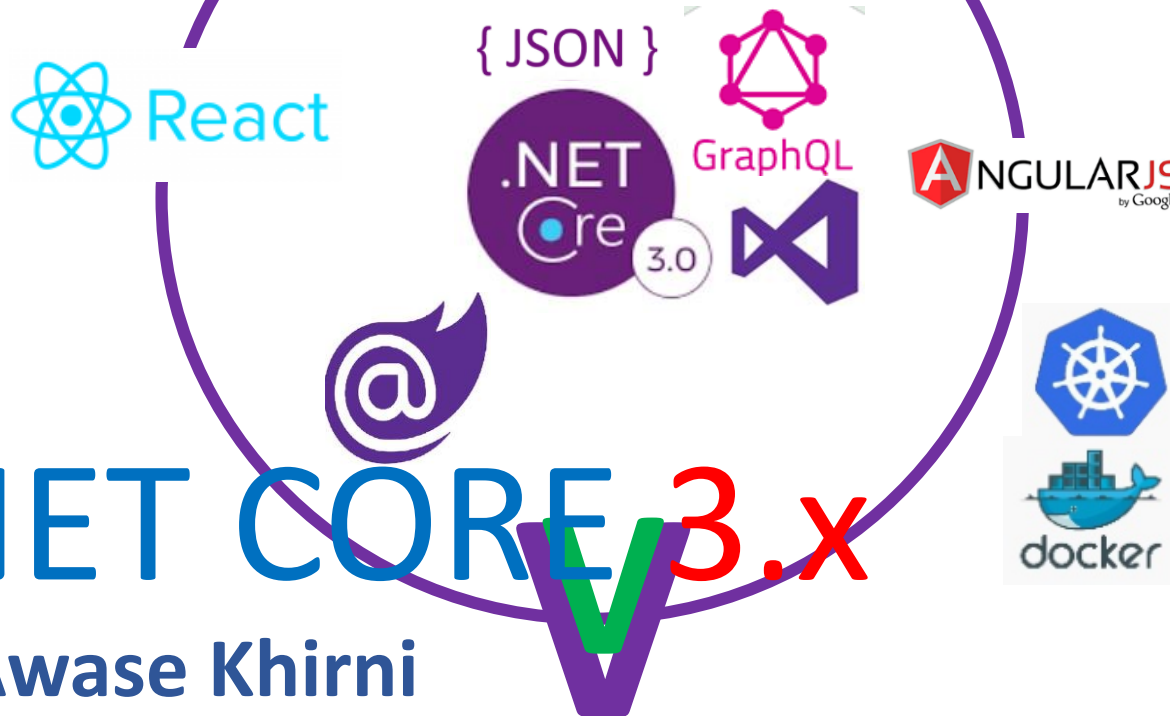
ASP.NET CORE 3.x

Syed Awase Khirni

RESEARCHER | ENTREPRENEUR | TECHNOLOGY COACH

@sak008 | sak@sycliq.com / sak@territorialprescience.com | +91. 9035433124

Syed Awase earned his PhD from University of Zurich in GIS, supported by EU V Framework Scholarship from SPIRIT Project (www.geo-spirit.org). He currently provides consulting services through his startup www.territorialprescience.com and www.sycliq.com. He empowers the ecosystem by sharing his technical skills worldwide, since 2008. He provides training in Java Technology Stack, .Net Technology Stack, R, DataScience, Client Side frameworks (Angular, KnockOut, Aurelia, Vue, Ember, Backbone etc..), Node Stack, Machine Learning, Python Stack, Php Stack. Elain Technologies Canada holds copyrights of the material and codeplay source code.



.NET Core Release History

Version	Release Date	Support	Support End Date	
.Net Core 1.0	2016/06/27		2019/06/27	
.Net Core 1.1	2016/11/16		2019/06/27	
.Net Core 2.0	2017/08/14		2018/10/1	
.Net Core 2.1	2018/05/30		2019/08/21	
.Net Core 2.2	2018/12/04		2019/12/23	
.Net Core 3.0	2019/09/23		2020/03/03	
.Net Core 3.1	2019/12/03			
.Net Core 3.2				

.NET Framework 4.8 Setup

- <https://blogs.windows.com/windowsexperience/2019/05/21/how-to-get-the-windows-10-may-2019-update/#o6c6CPgiYoSk8rGK.97>
- <http://go.microsoft.com/fwlink/?LinkId=2085155>

OS Platform	.NET Framework 4.8 Redistributable	.NET Framework 4.8 Language Pack
Windows 7 SP1/Windows Server 2008 R2	KB4503548	KB4497410
Windows Server 2012	KB4486081	KB4087513
Windows 8.1/Windows Server 2012 R2	KB4486105	KB4087514
Windows 10 Version 1607	KB4486129 (Catalog Only)	KB4087515 (Catalog Only)
Windows 10 Version 1703	KB4486129	KB4087515
Windows Server 2016	KB4486129 (Catalog Only)	KB4087515 (Catalog Only)
Windows 10 Version 1709	KB4486153	KB4087642
Windows 10 Version 1803	KB4486153	KB4087642
Windows Server, version 1803	KB4486153	KB4087642
Windows 10 Version 1809	KB4486153	KB4087642
Windows Server, version 1809	KB4486153 (Catalog Only)	KB4087642 (Catalog Only)
Windows Server 2019	KB4486153 (Catalog Only)	KB4087642 (Catalog Only)

.NET Framework 4.8

- Included in Windows 10 May 2019 update
- Also available on
 - Windows 7+
 - Windows Server 2008 R2+.
- Runtime
 - JIT improvements
 - NGEN improvements
 - Antimalware Scanning for All Assemblies
- BCL
 - Updated Zlib
 - Reducing FIPS impact on Cryptography
- Windows Forms
 - Accessibility Enhancements
 - UIA LiveRegions Support in Labels and StatusStrips
 - UIA Notification Events
 - ToolTips on keyboard access
 - DataGridView control accessible hierarchy changes
- WCF
 - ServiceHealthBehaviour
- WPF
 - Screen narrators no longer announce collapse
 - High DPI enhancements
 - Tooltips on keyboard accesss
 - Added Support for SizeOfSet and PositionInSet Automation Properties

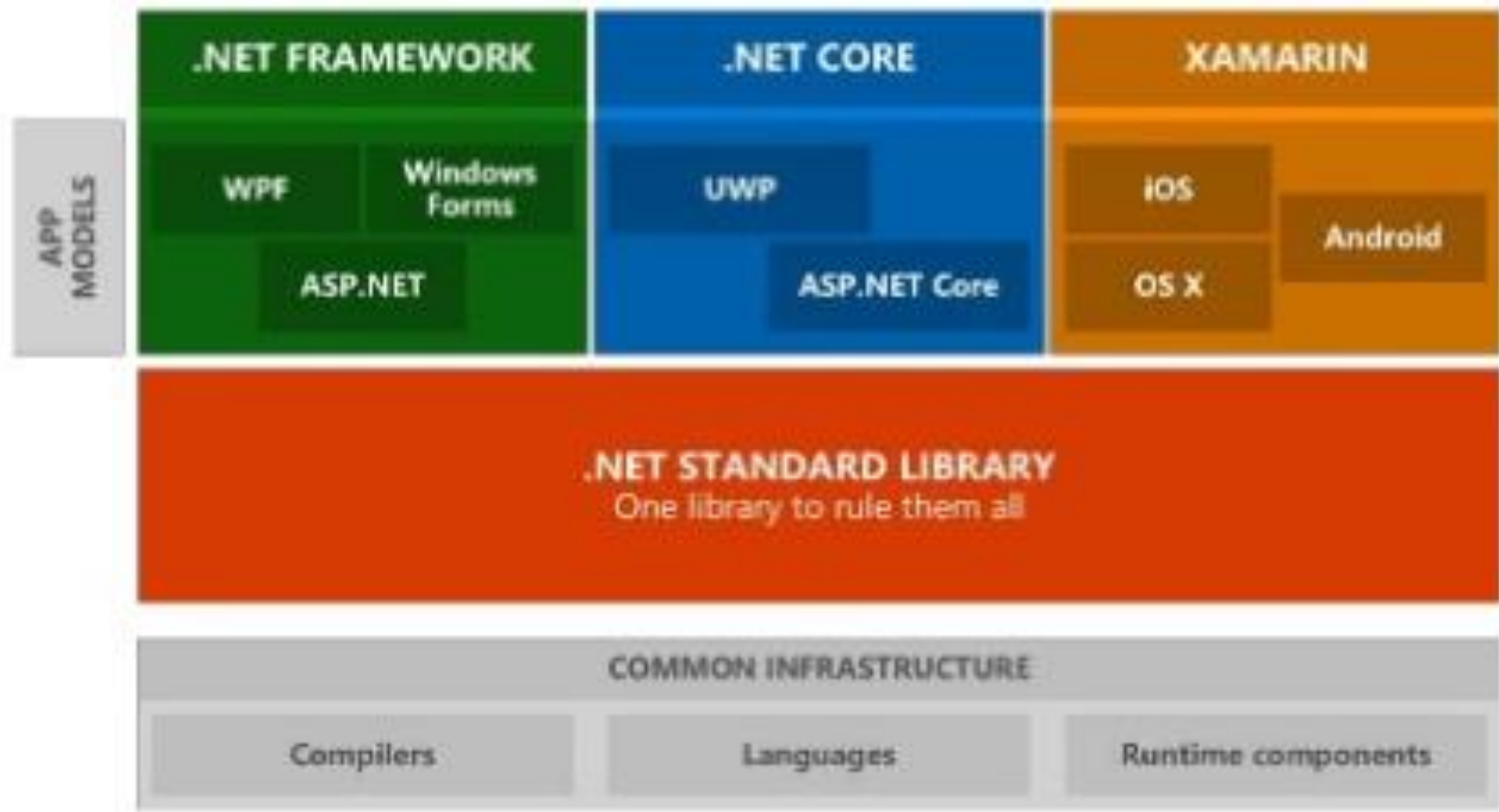
.NET Core 3.x

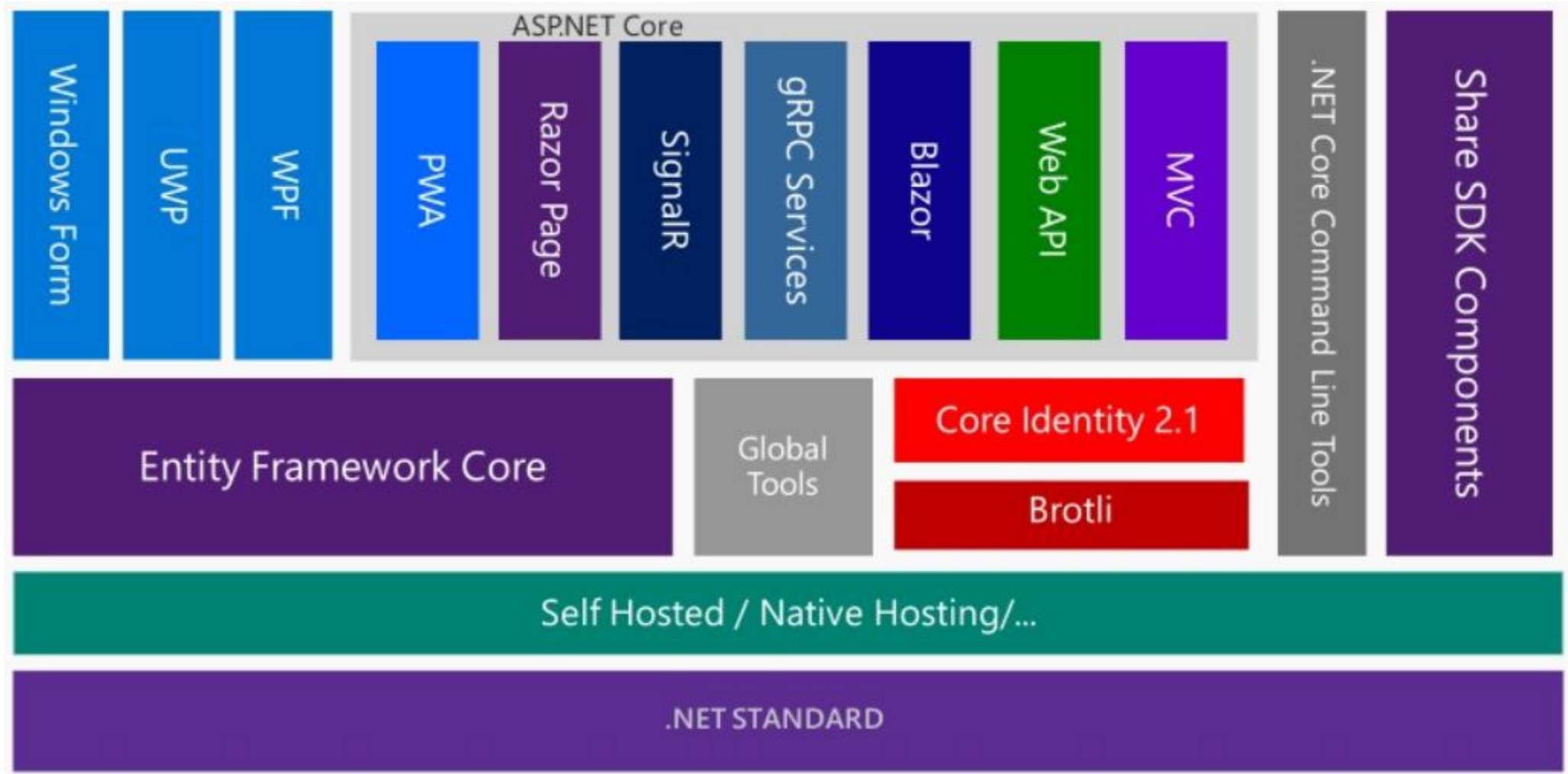
- Free and open source
- C#, F#
- Cross platform support
 - Windows
 - macOS
 - linux

ASP.NET	API, MVC, Razor Pages, Blazor Server, Blazor client, gRPC, SignalR
Windows Forms	
WPF	
Entity Framework	.Net Standard 2.1, x-plat lightweight, C#8 (Async streams, nullable reference types) CosmosDB
WCF	Core WCF, Rebuild to gRPC

Current Platform Requirements

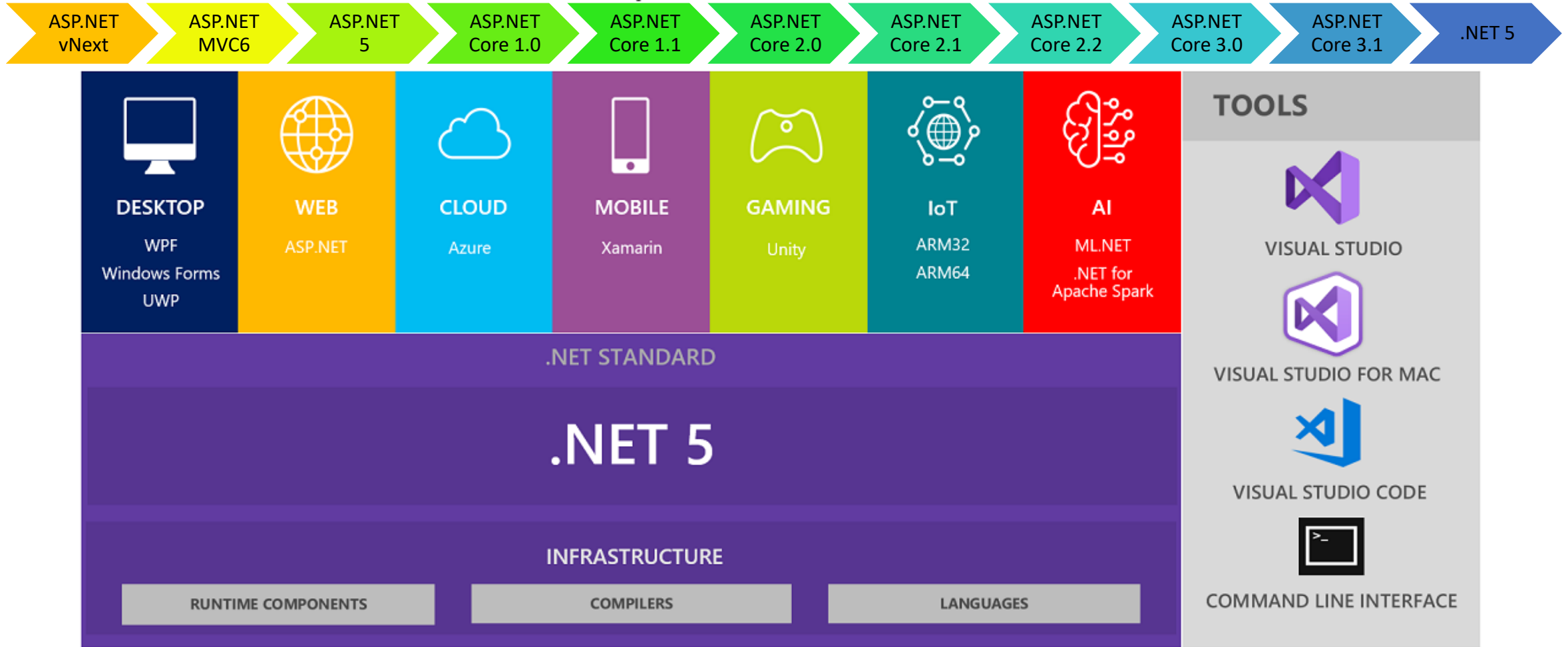
- High Performance Framework/Platform, Highly Secure and Easily Scalable platform.
- Efficient memory management.
- Cross-platform support with CI/CD – develop once deploy everywhere.
- Support for Machine Learning and AI
- Server Support for HTTP/S + ALL FEATURES
- Encrypted HTTPS connection through SSL
- Browser Support for HTTP/2
- Easy Caching and Identity Management
- High performance RPC services
- Support for WebAssembly.





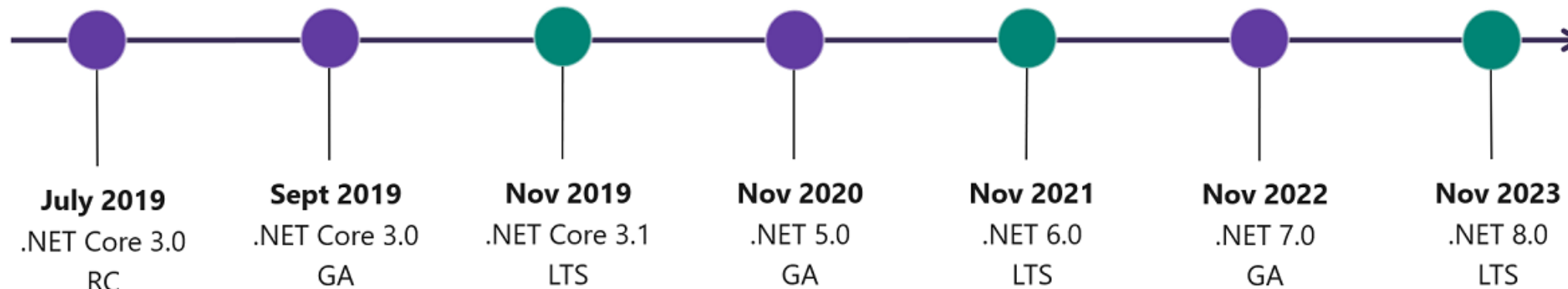
(source: Microsoft)

.NET – A unified platform



(source: Microsoft)

.NET Schedule



- .NET Core 3.0 release in September
- .NET Core 3.1 = Long Term Support (LTS)
- .NET 5.0 release in November 2020
- Major releases every year, LTS for even numbered releases
- Predictable schedule, minor releases if needed

(source: Microsoft)

Download .NET Core

.NET Core is a cross-platform version of .NET for building websites, services, and console apps.

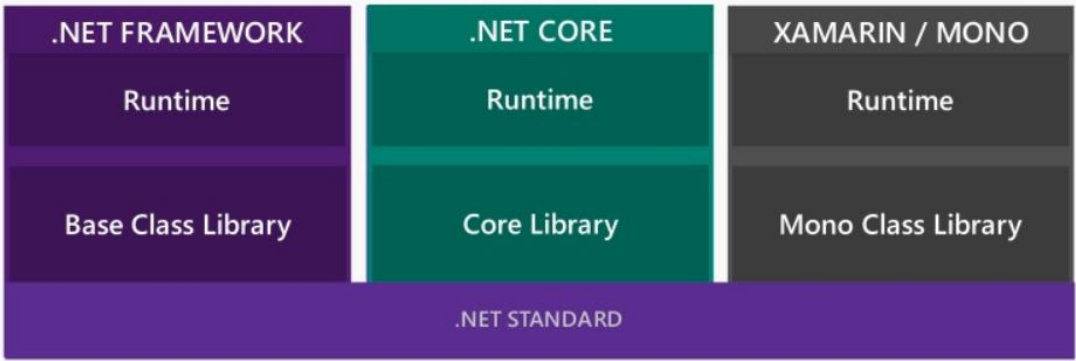
? Not sure what to download? [See recommended downloads for the latest version of .NET.](#)

Version	Status	Latest release	Latest release date	End of support
.NET 5.0	Preview ⓘ	5.0.0-preview.7	2020-07-21	
.NET Core 3.1 (recommended)	LTS ⓘ	3.1.7	2020-08-11	2022-12-03
.NET Core 3.0	End of life ⓘ	3.0.3	2020-02-18	2020-03-03
.NET Core 2.2	End of life ⓘ	2.2.8	2019-11-19	2019-12-23
.NET Core 2.1	LTS ⓘ	2.1.21	2020-08-11	2021-08-21
.NET Core 2.0	End of life ⓘ	2.0.9	2018-07-10	2018-10-01
.NET Core 1.1	End of life ⓘ	1.1.13	2019-05-14	2019-06-27
.NET Core 1.0	End of life ⓘ	1.0.16	2019-05-14	2019-06-27

Source: <https://dotnet.microsoft.com/download/dotnet-core>

.NET 5

.NET



.NET5



ASP.NET CORE 3.0

- .NET Core 3.0 implements .NET standard 2.1
- .NET Core 3.0 adds support for C# 8.0
 - Requires Visual Studio 2019 version 16.3 or newer version.
 - VSCode with latest C# Extension
- New JSON Serializer
 - Uses System.Text.Json by default for JSON serialization
 - Reads and writes JSON asynchronously.
 - Optimized for UTF-8 text.
 - Higher performance than Newtonsoft.JSON
- Worker Service : used for long running tasks such as Windows services or linux Systemd daemons.
- Single EXEs publish
 - Dotnet publish -r win10-x64 /p:PublishSingleFile=true.
- Angular template updated to Angular 8.
- Performance improvements
 - Less memory consumption for using the built-in dependency injection container for scoped services
 - Less memory consumption for websocket connections.
 - Memory reduction and throughput improvements for HTTPS connections
- A high performance open source universal RPC framework – gRPC
- Blazor- a framework for building interactive client-side web UI with .NET
- SignalR now uses System.Text.Json to serialize and deserialize JSON messages
- New SqlClient- System.Data.SqlClient

Setting up ASP.NET Core 3.x

<https://dotnet.microsoft.com/download/dotnet/current>

SYED AWASE KHIRNI

Download .NET Core 3.1

- <https://download.visualstudio.microsoft.com/download/pr/e24a5908-d199-4575-b94b-5828bbbc956e/bc1b7a8c4e06a18780eeb15925aaee9b/dotnet-sdk-3.1.107-win-x64.exe>

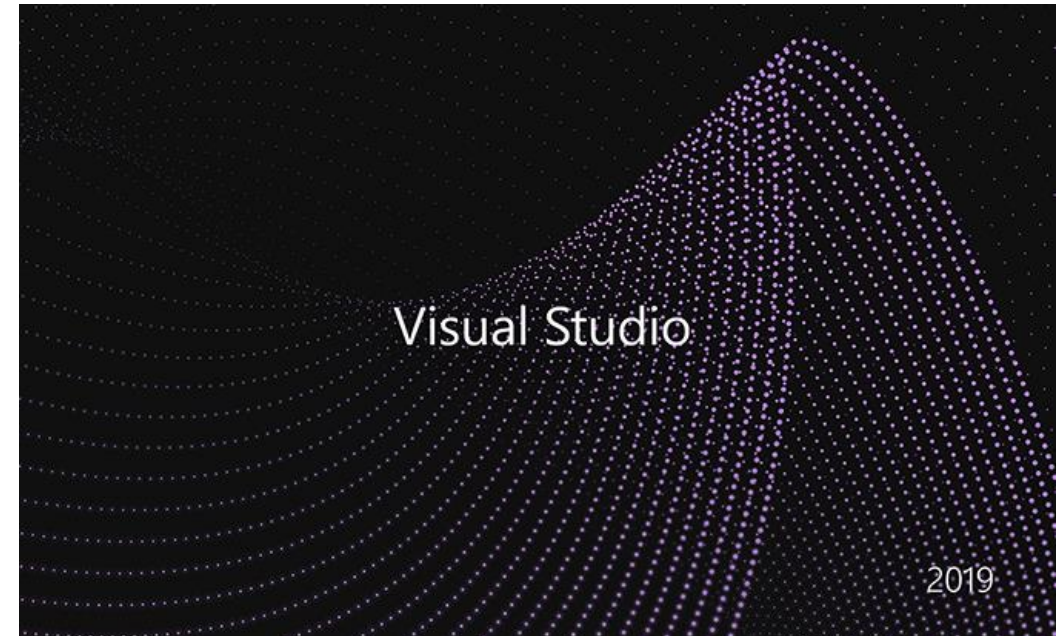
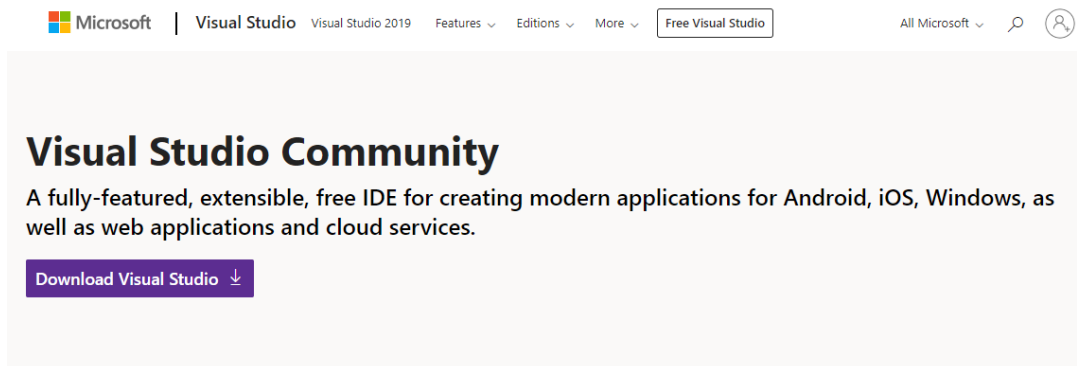
SDK 3.1.107
Visual Studio support
Visual Studio 2019 (v16.4)
Included in
Visual Studio 16.4.12
Included runtimes
.NET Runtime 3.1.7
ASP.NET Core Runtime 3.1.7
Desktop Runtime 3.1.7
Language support
C# 8.0
F# 4.7

OS	Installers	Binaries
Linux	Package manager instructions	ARM32 ARM64 x64 Alpine x64 RHEL 6 x64
macOS	x64	x64
Windows	x64 x86	ARM32 x64 x86
All	dotnet-install scripts	

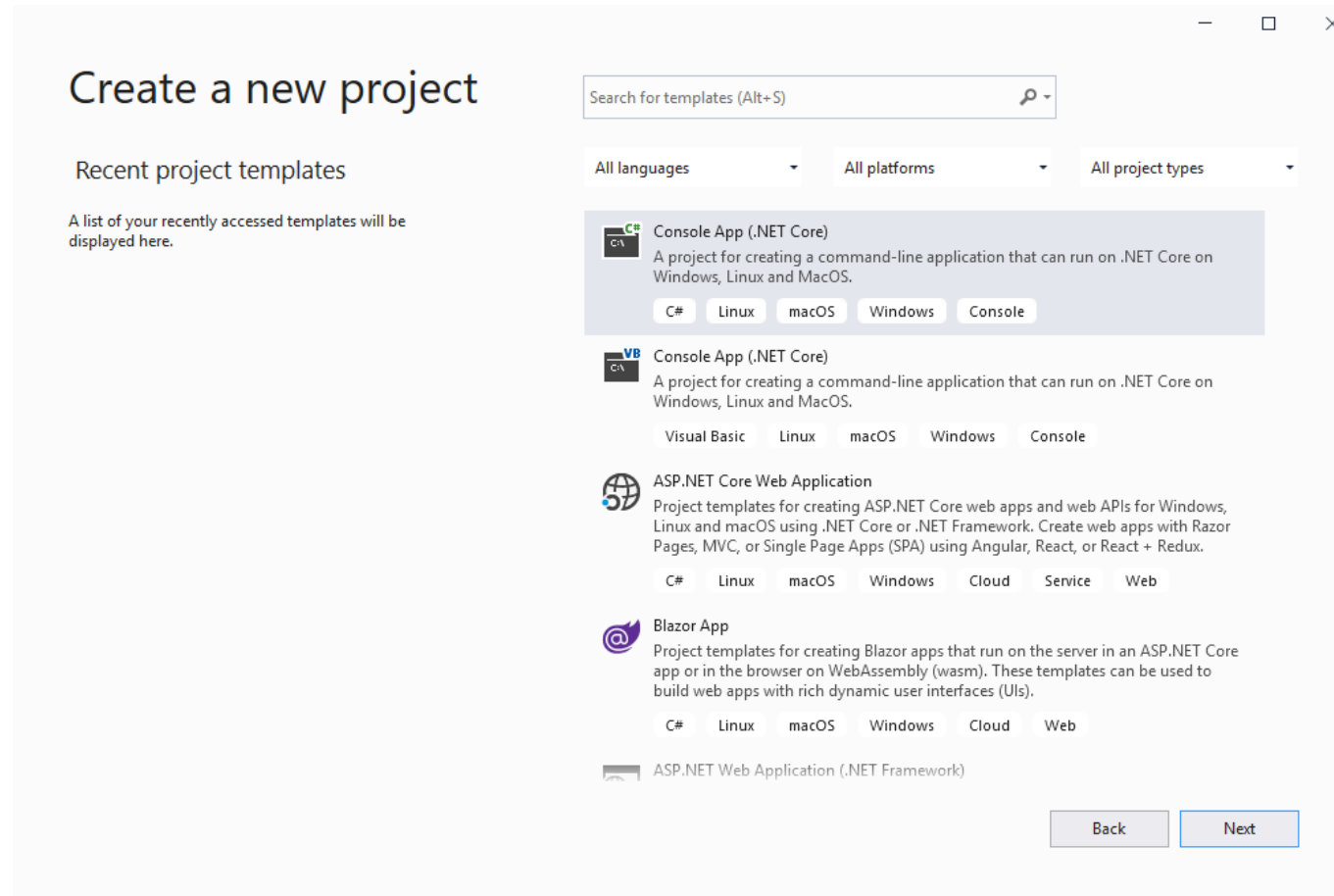
<https://dotnet.microsoft.com/download/dotnet/current>

Visual Studio Community IDE 2019

- <https://visualstudio.microsoft.com/vs/community/>
- .NET Core 3.1 works with Visual Studio Community IDE 2019 and above only.



Visual Studio 2019 Community Edition



SYED AWASE KHIRNI

Example 1: ConsoleApp .NET CORE

Learning Outcomes:

1. **Creating a .Net Core 3.x console application using visual studio 2019.**

Create a new project


Recent project templates


A list of your recently accessed templates will be displayed here.


SELECT Console App =>


Search for templates (Alt+S) 🔍


All languages ▾ All platforms ▾ All project types ▾

 **Console App (.NET Core)**
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Console

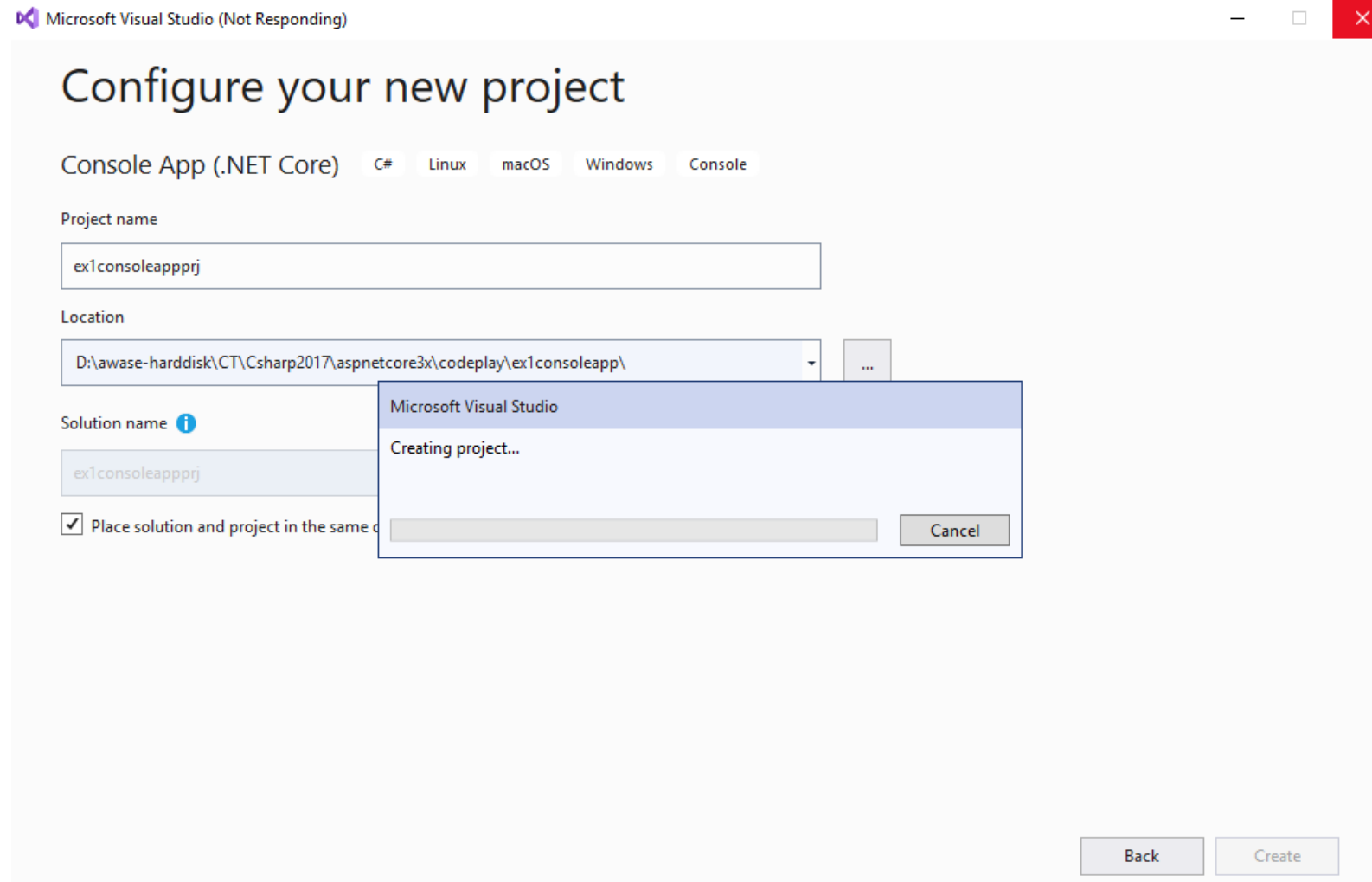
 **Console App (.NET Core)**
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
Visual Basic Linux macOS Windows Console

 **ASP.NET Core Web Application**
Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.
C# Linux macOS Windows Cloud Service Web

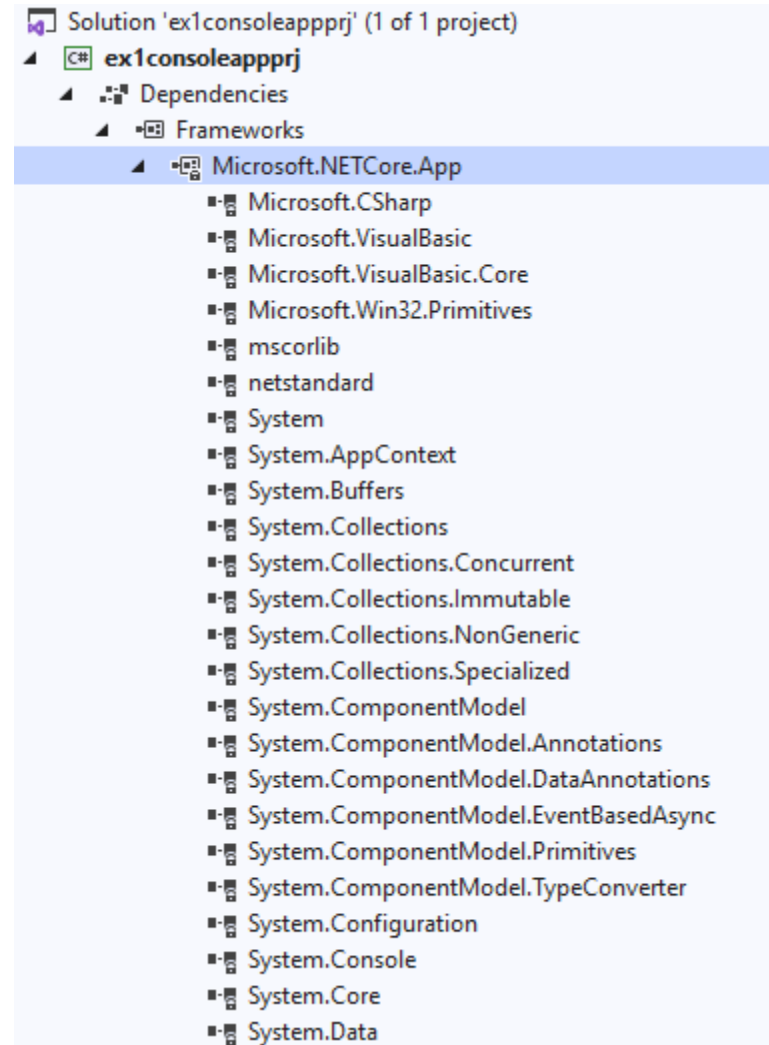
 **Blazor App**
Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).
C# Linux macOS Windows Cloud Web

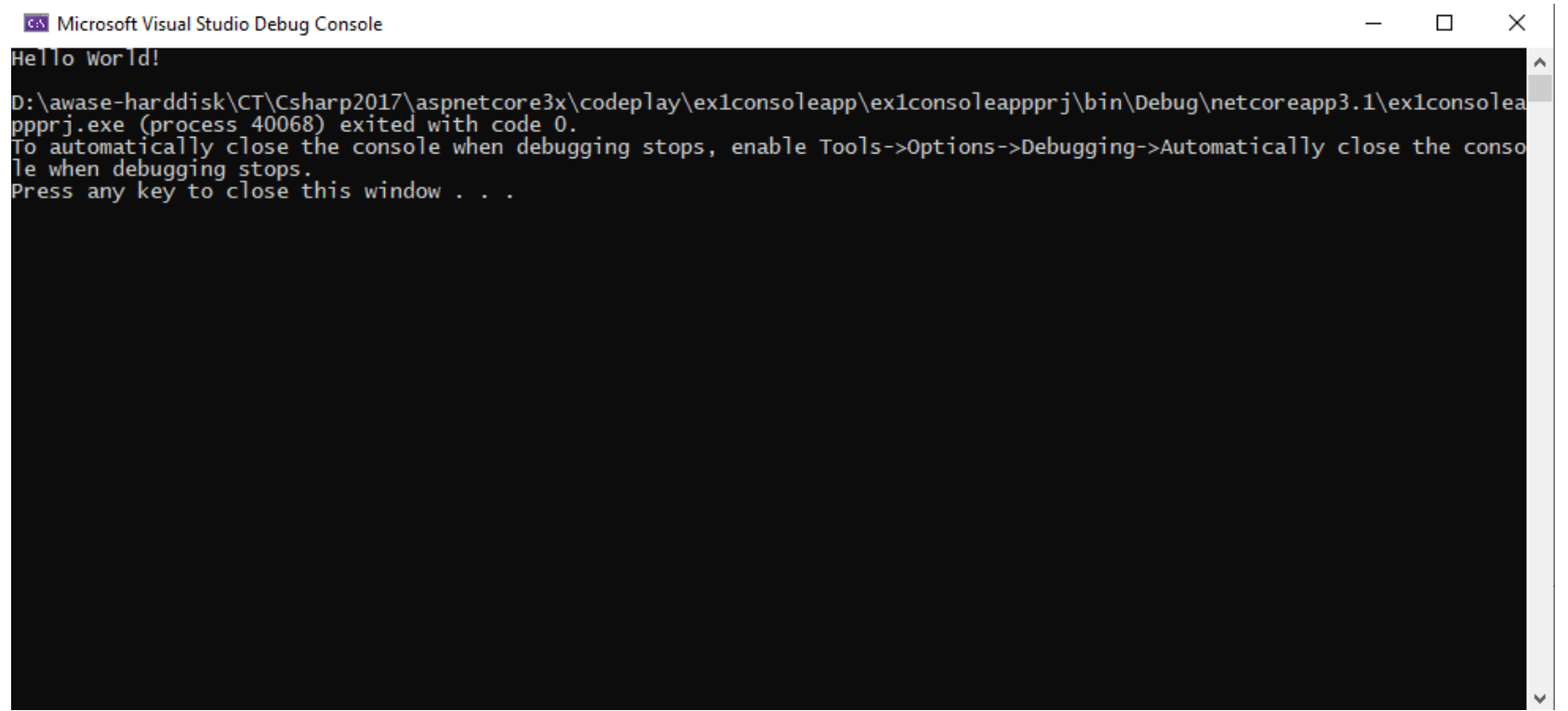
 **ASP.NET Web Application (.NET Framework)**

Back Next



```
1 using System;
2
3 namespace ex1consoleappprj
4 {
5     -references
6     class Program
7     {
8         -references
9         static void Main(string[] args)
10        {
11            Console.WriteLine("Hello World!");
12        }
13    }
14 }
```





```
Microsoft Visual Studio Debug Console

Hello World!
D:\awase-harddisk\CT\Csharp2017\aspnetcore3x\codeplay\ex1consoleapp\ex1consoleappprj\bin\Debug\netcoreapp3.1\ex1consoleappprj.exe (process 40068) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

SYED AWASE KHIRNI

Example 2: ASP.NET Core Web Application

Learning Outcomes:

1. Understand the steps to create ASP.NET Core 3.x empty web application
2. Understand the web application structure
3. Understand the key dependencies for ASP.NET Core 3.x empty web application
4. Executing ASP.NET Core 3.x empty web application and rendering it on <http://localhost:44346>

Create a new project

Recent project templates

Console App (.NET Core)

C#

All languages

All platforms

All project types

Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

C#

Linux

macOS

Windows

Console

Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

Visual Basic

Linux

macOS

Windows

Console

ASP.NET Core Web Application

Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and MacOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

C#

Linux

macOS

Windows

Cloud

Service

Web

Blazor App

Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).

C#

Linux

macOS

Windows

Cloud

Web

ASP.NET Web Application (.NET Framework)

Next

SELECT TO CREATE

Configure your new project

ASP.NET Core Web Application C# Linux macOS Windows Cloud Service Web

Project name
ex2emptyaspnetcoreprj

Location
D:\awase-harddisk\CT\Csharp2017\aspnetcore3x\codeplay\ex2emptyaspnetcore\

Solution
Create new solution

Solution name ⓘ
ex2emptyaspnetcoreprj

☒ Place solution and project in the same directory


Back


Create


Create a new ASP.NET Core web application


.NET Core


ASP.NET Core 3.1


**Empty**
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

**API**
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

**Web Application**
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

**Web Application (Model-View-Controller)**
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

**Angular**
A project template for creating an ASP.NET Core application with Angular

**React.js**

Authentication
No Authentication
[Change](#)

Advanced
☒ Configure for HTTPS
☐ Enable Docker Support
(Requires [Docker Desktop](#))

Linux

Author: Microsoft
Source: Templates 3.1.8

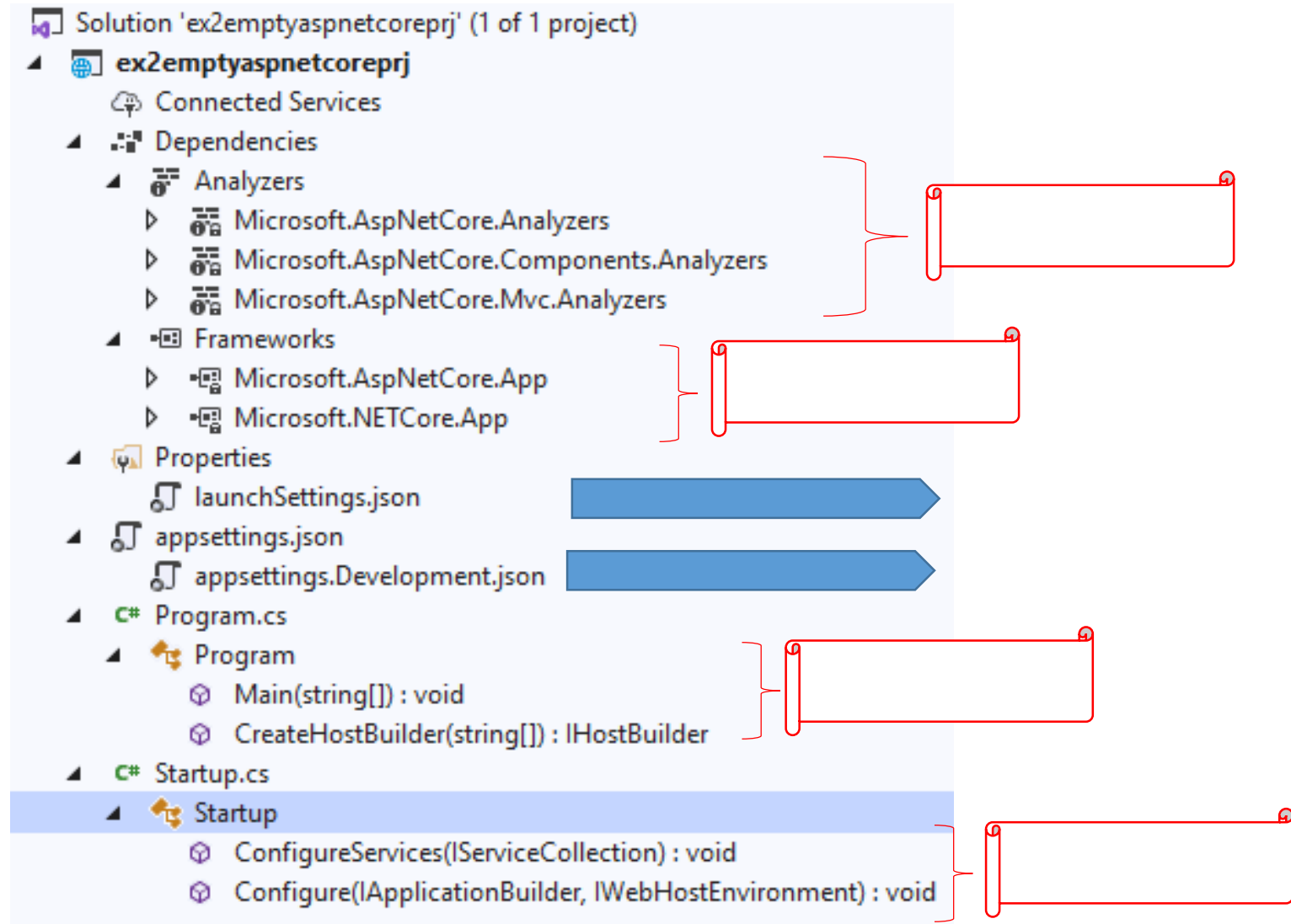
Back

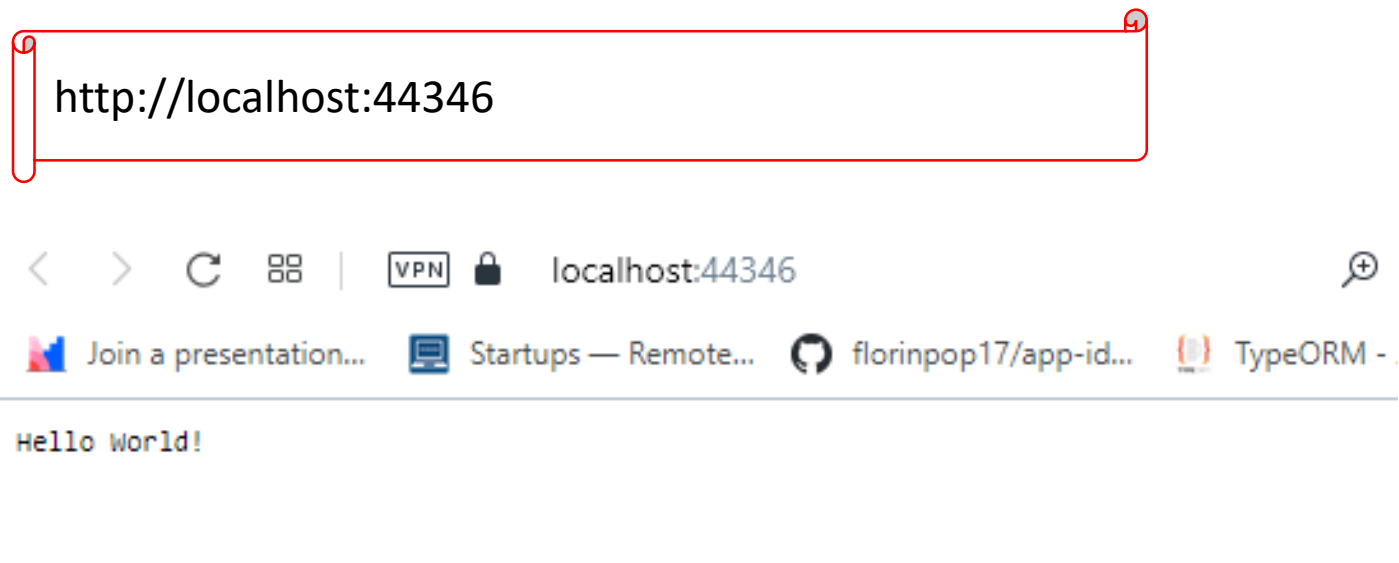
Create

[Get additional project templates](#)

The screenshot displays the Visual Studio IDE with the 'Startup.cs' file open in the editor. The code defines a 'Startup' class with a 'ConfigureServices' method. The Solution Explorer on the right shows the project structure, and the Output window at the bottom is set to 'Debug' mode.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using Microsoft.AspNetCore.Builder;
6 using Microsoft.AspNetCore.Hosting;
7 using Microsoft.AspNetCore.Http;
8 using Microsoft.Extensions.DependencyInjection;
9 using Microsoft.Extensions.Hosting;
10
11 namespace ex2emptyaspnetcoreprj
12 {
13     public class Startup
14     {
15         // This method gets called by the runtime. Use this method to add services to the container.
16         // For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?Li
17         public void ConfigureServices(IServiceCollection services)
18         {
19         }
20
21         // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
```





SYED AWASE KHIRNI

Example 3: ASP.NET Core Razor Web Application

Learning Outcomes:

1. Understand the steps to create ASP.NET Core 3.x web application with razor pages
2. Understand the web application structure
3. Understand the key dependencies for ASP.NET Core 3.x web application with razor pages
4. Executing ASP.NET Core 3.x web application and rendering it on <http://localhost:44360>

Configure your new project

ASP.NET Core Web Application

C#

Linux

macOS

Windows

Cloud

Service

Web

Project name

ex3aspnetcorerazorprj


Location

D:\awase-harddisk\CT\Csharp2017\aspnetcore3x\codeplay\ex3aspnetcorerazor\

...

Solution

Create new solution

Solution name 

ex3aspnetcorerazorprj

☒ Place solution and project in the same directory

Back

Create

Create a new project

Recent project templates

 Console App (.NET Core) C#

Search for templates (Alt+S)

All languages

All platforms

All project types



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

C#

Linux

macOS

Windows

Console



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

Visual Basic

Linux

macOS

Windows

Console



ASP.NET Core Web Application

Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

C#

Linux

macOS

Windows

Cloud

Service

Web



Blazor App

Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).

C#

Linux

macOS

Windows

Cloud

Web



ASP.NET Web Application (.NET Framework)


Next

SELECT TO CREATE


Create a new ASP.NET Core web application

.NET Core


ASP.NET Core 3.1

 **Empty**


An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

 **API**


A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

 **Web Application**


A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

 **Web Application (Model-View-Controller)**

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

 **Angular**

A project template for creating an ASP.NET Core application with Angular

 **React.js**

Get additional project templates

Authentication

No Authentication

[Change](#)

Advanced

☒ Configure for HTTPS

☐ Enable Docker Support
(Requires [Docker Desktop](#))

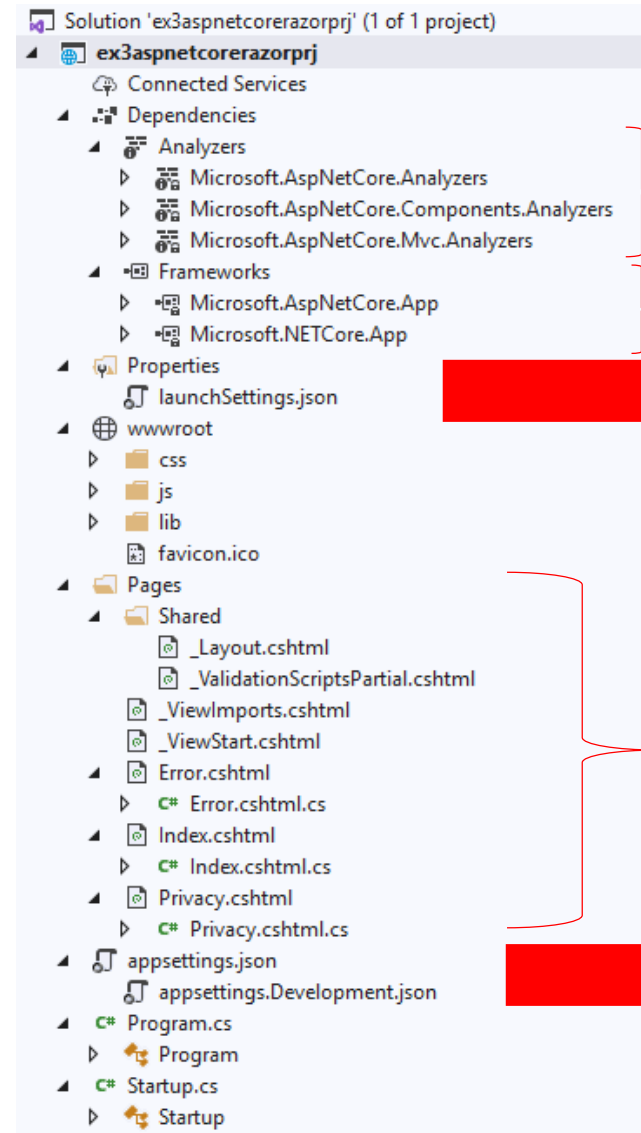
Linux

☐ Enable Razor runtime compilation

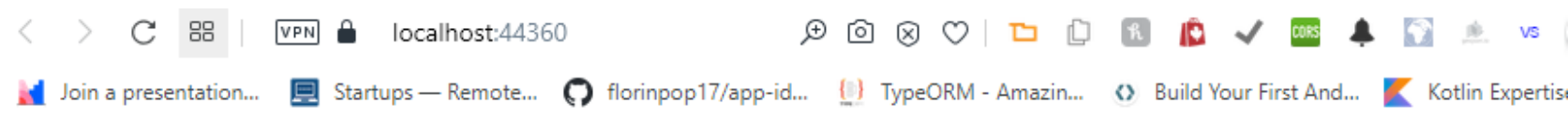
Author: Microsoft
Source: Templates 3.1.8

Back

Create



http://localhost:44360



ex3aspnetcorerazorprj Home Privacy

Welcome

Learn about [building Web apps with ASP.NET Core](#).

SYED AWASE KHIRNI


Example 4:ASP.NET Core Web Api Project

Learning Outcomes:

1. Understand the steps to create ASP.NET Core 3.x web api project
2. Understand the web application structure
3. Understand the key dependencies for ASP.NET Core 3.x web api project
4. Executing ASP.NET Core 3.x web api and rendering it on <http://localhost:44360>

Create a new project

Recent project templates

-  ASP.NET Core Web Application C#
-  Console App (.NET Core) C#

Search for templates (Alt+S)

All languages

All platforms

All project types



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

C# Linux macOS Windows Console



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

Visual Basic Linux macOS Windows Console



ASP.NET Core Web Application

Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

C# Linux macOS Windows Cloud Service Web



Blazor App

Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).

C# Linux macOS Windows Cloud Web



ASP.NET Web Application (.NET Framework)

Next

SELECT TO CREATE

Configure your new project

ASP.NET Core Web Application

C#

Linux

macOS

Windows

Cloud

Service

Web

Project name

ex4aspnetwebapi

Location

D:\awase-harddisk\CT\Csharp2017\aspnetcore3x\codeplay\ex4aspnetwebapi\

...

Solution

Create new solution

Solution name ⓘ

ex4aspnetwebapi

☒ Place solution and project in the same directory

Back

Create

Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 3.1

Empty
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

Web Application
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

Web Application (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

Angular
A project template for creating an ASP.NET Core application with Angular

React.js

[Get additional project templates](#)

Authentication

No Authentication

[Change](#)

Advanced

☒ Configure for HTTPS

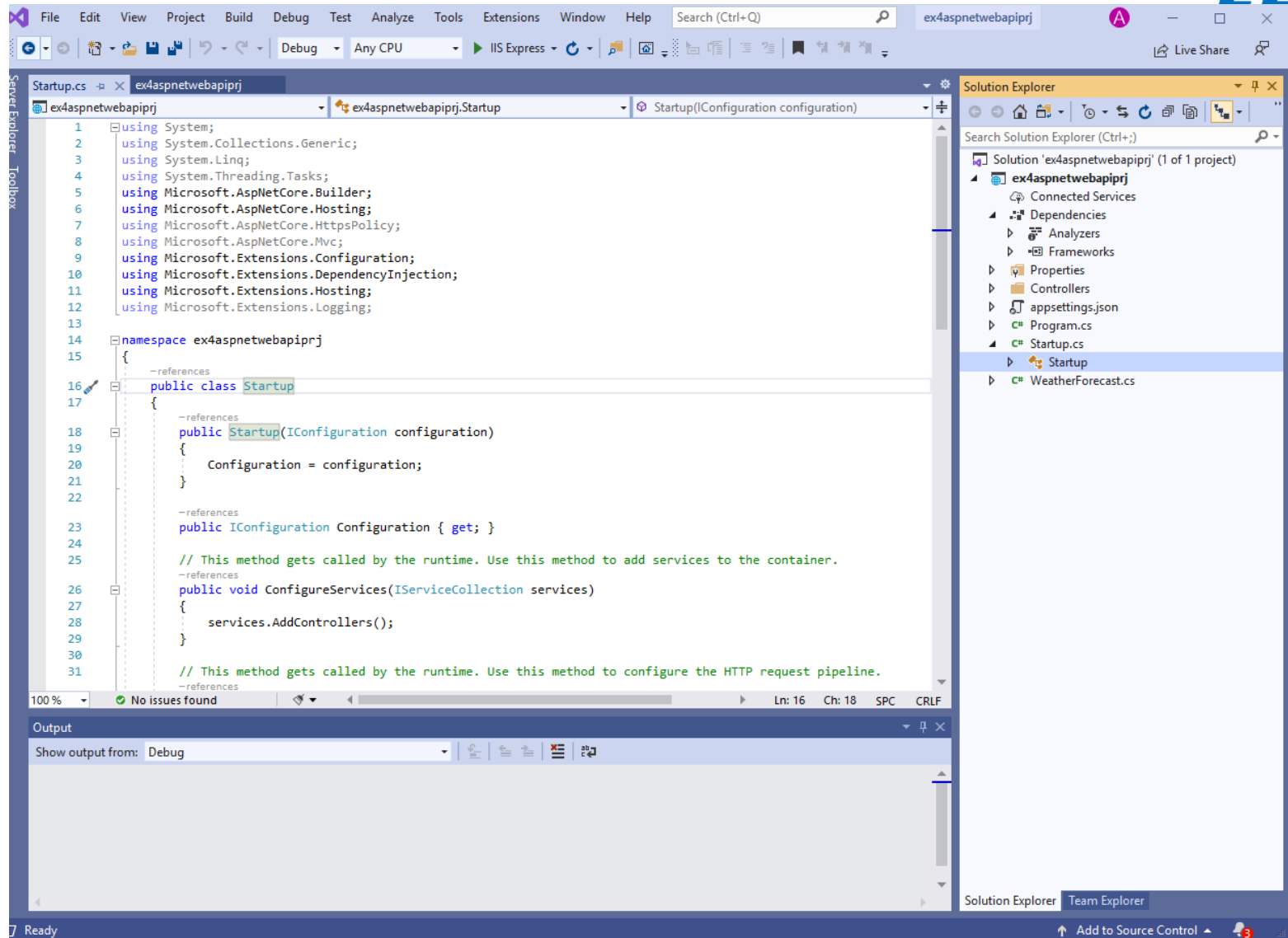
☐ Enable Docker Support
(Requires [Docker Desktop](#))

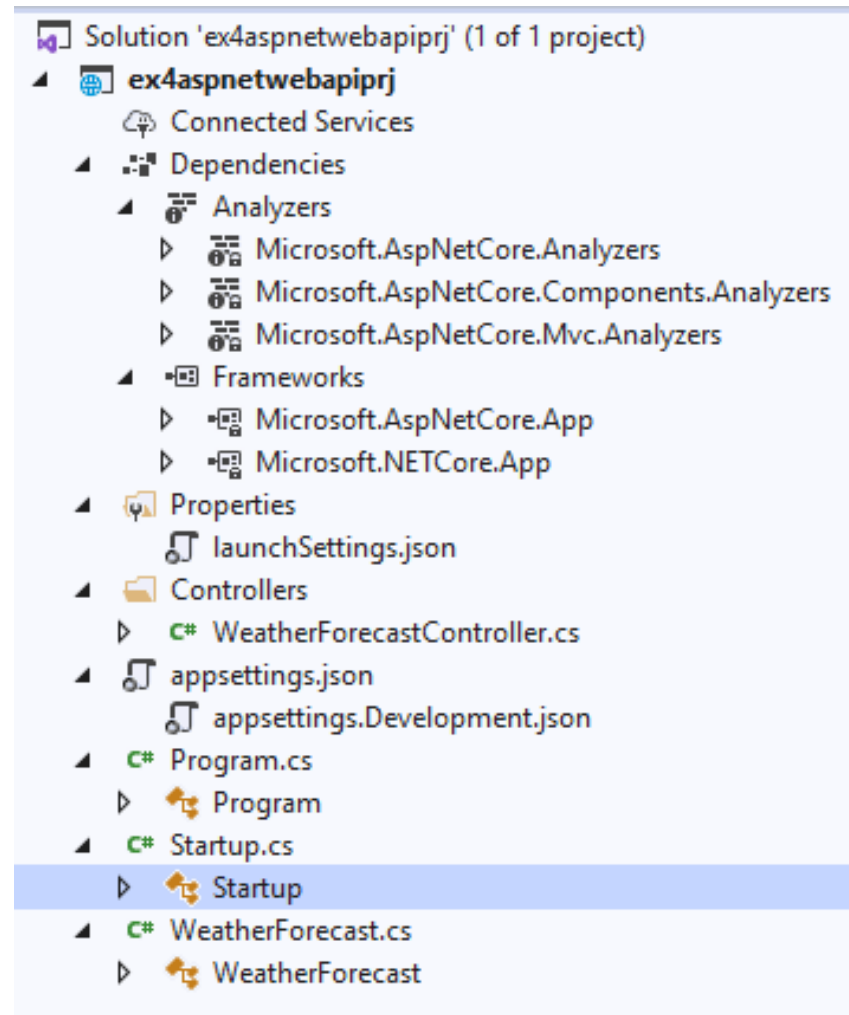
Linux

Author: Microsoft
Source: Templates 3.1.8

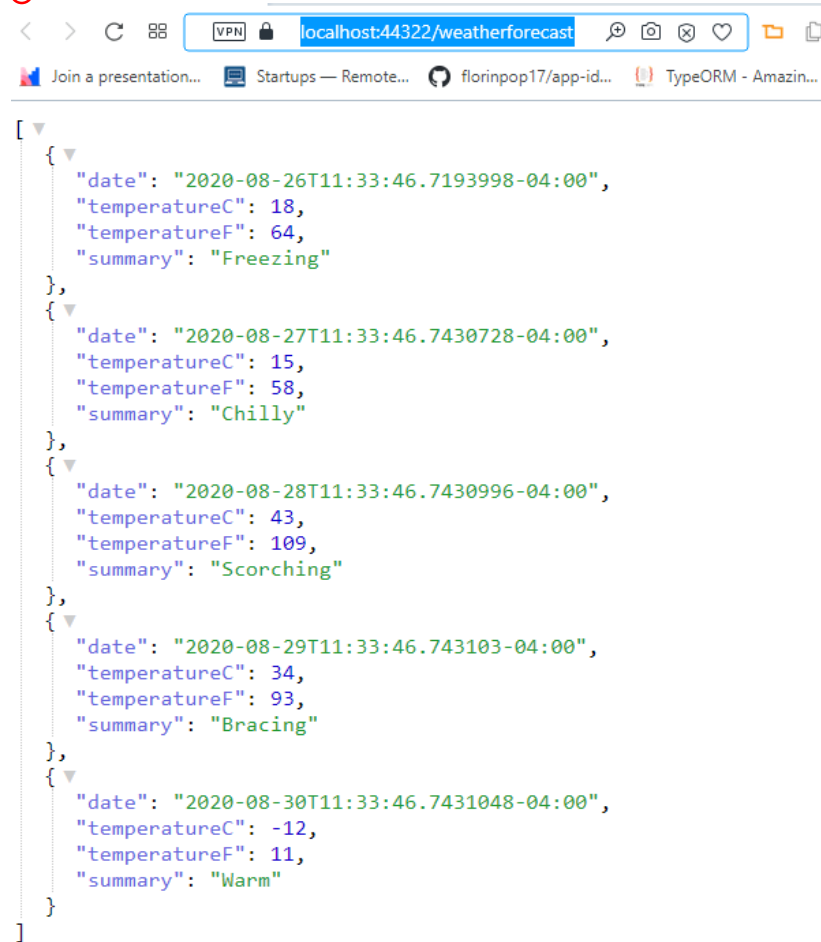
Back

Create





http://localhost:44322/weatherforecast



```
[
  {
    "date": "2020-08-26T11:33:46.7193998-04:00",
    "temperatureC": 18,
    "temperatureF": 64,
    "summary": "Freezing"
  },
  {
    "date": "2020-08-27T11:33:46.7430728-04:00",
    "temperatureC": 15,
    "temperatureF": 58,
    "summary": "Chilly"
  },
  {
    "date": "2020-08-28T11:33:46.7430996-04:00",
    "temperatureC": 43,
    "temperatureF": 109,
    "summary": "Scorching"
  },
  {
    "date": "2020-08-29T11:33:46.743103-04:00",
    "temperatureC": 34,
    "temperatureF": 93,
    "summary": "Bracing"
  },
  {
    "date": "2020-08-30T11:33:46.7431048-04:00",
    "temperatureC": -12,
    "temperatureF": 11,
    "summary": "Warm"
  }
]
```

SYED AWASE KHIRNI

Example 5:ASP.NET Core MVC Web Application

Learning Outcomes:

1. Understand the steps to create ASP.NET Core 3.x mvc web application
2. Understand the web application structure
3. Understand the key dependencies for ASP.NET Core 3.x mvc web application
4. Executing ASP.NET Core 3.x mvc web application and rendering it on <http://localhost:44360>

Create a new project

Recent project templates

ASP.NET Core Web Application C#

Console App (.NET Core) C#

Search for templates (Alt+S)

All languages All platforms All project types

Console App (.NET Core)
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Console

Console App (.NET Core)
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
Visual Basic Linux macOS Windows Console

ASP.NET Core Web Application
Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.
C# Linux macOS Windows Cloud Service Web

Blazor App
Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).
C# Linux macOS Windows Cloud Web

ASP.NET Web Application (.NET Framework)

SELECT TO CREATE

Next

Updated on May 2020

8/25/2020

Copyright 2020 www.elain-tech.com Syed Awase Khirni

45

Configure your new project

ASP.NET Core Web Application

C#

Linux

macOS

Windows

Cloud

Service

Web

Project name

ex5aspnetcoremvcprj


Location

D:\awase-hddisk\CT\Csharp2017\aspnetcore3x\codeplay\ex5apsnetcoremvc\

...

Solution

Create new solution

Solution name 

ex5aspnetcoremvcprj

☒ Place solution and project in the same directory


Back

Create


Create a new ASP.NET Core web application

.NET Core


ASP.NET Core 3.1

 **Empty**


An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

 **API**


A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

 **Web Application**


A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

 **Web Application (Model-View-Controller)**

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

 **Angular**

A project template for creating an ASP.NET Core application with Angular

 **React.js**

Get additional project templates

Authentication

No Authentication

[Change](#)

Advanced

☒ Configure for HTTPS

☐ Enable Docker Support
(Requires [Docker Desktop](#))

Linux

☐ Enable Razor runtime compilation

Author: Microsoft
Source: Templates 3.1.8

Back

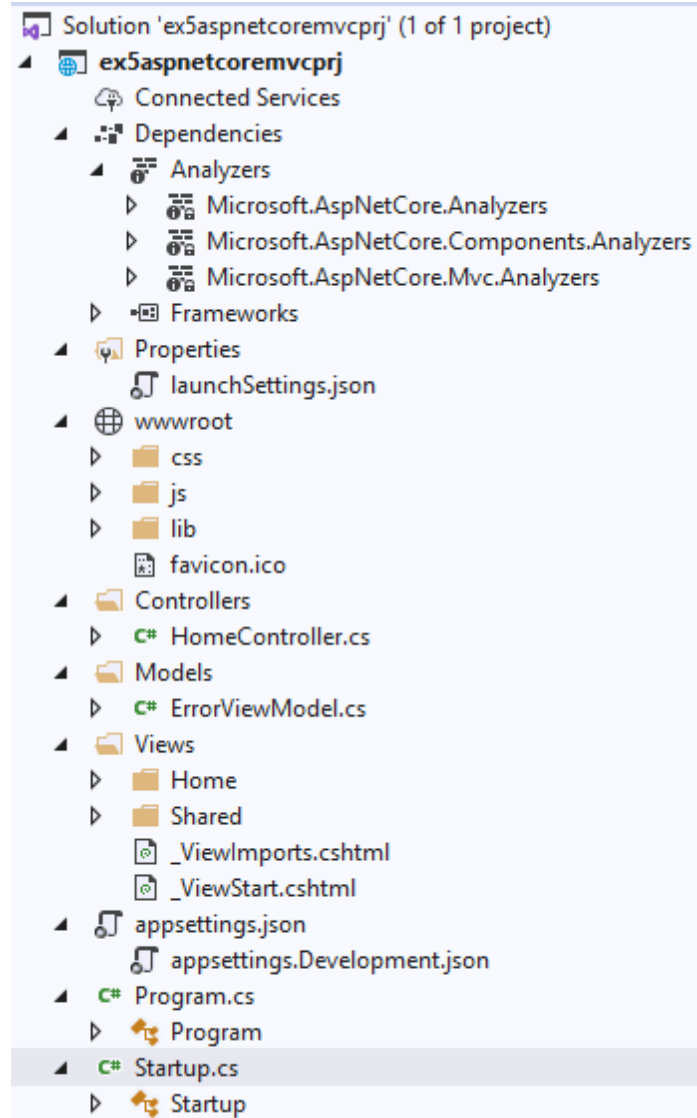
Create

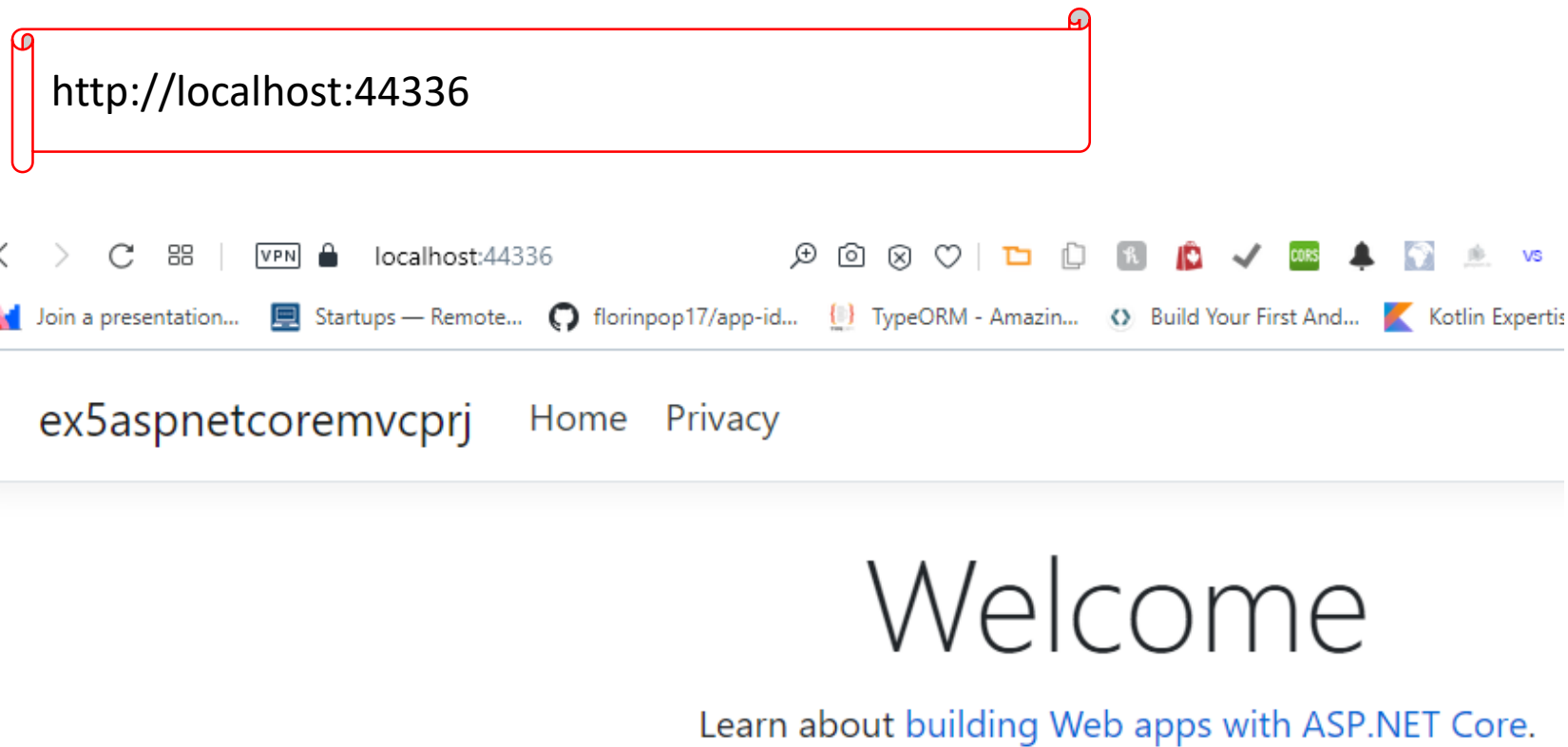
```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using Microsoft.AspNetCore.Builder;
6  using Microsoft.AspNetCore.Hosting;
7  using Microsoft.AspNetCore.HttpsPolicy;
8  using Microsoft.Extensions.Configuration;
9  using Microsoft.Extensions.DependencyInjection;
10 using Microsoft.Extensions.Hosting;
11
12 namespace ex5aspnetcoremvcprj
13 {
14     2 references
15     public class Startup
16     {
17         0 references
18         public Startup(IConfiguration configuration)
19         {
20             Configuration = configuration;
21         }
22
23         1 reference
24         public IConfiguration Configuration { get; }
25
26         // This method gets called by the runtime. Use this method to add services to the container.
27         0 references
28         public void ConfigureServices(IServiceCollection services)
29         {
30             services.AddControllersWithViews();
31         }
32
33         // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
34         0 references
35         public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
36         {
37         }
38     }
39 }

```

The Solution Explorer on the right shows the project structure for 'ex5aspnetcoremvcprj', including Connected Services, Dependencies, Analyzers, Frameworks, Properties, wwwroot, Controllers, Models, Views, appsettings.json, Program.cs, and Startup.cs.





SYED AWASE KHIRNI


Example 6: ASP.NET Core Angular Stack Application


Learning Outcomes:

1. Understand the steps to create ASP.NET Core 3.x angular stack application
2. Understand the web application structure
3. Understand the key dependencies for ASP.NET Core 3.x angular stack application
4. Executing ASP.NET Core 3.x angular stack application and rendering it on <http://localhost:44360>

Create a new project

Recent project templates


 ASP.NET Core Web Application C#


 Console App (.NET Core) C#


All languages


All platforms


All project types

 Console App (.NET Core)
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Console

 Console App (.NET Core)
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
Visual Basic Linux macOS Windows Console

 ASP.NET Core Web Application
Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.
C# Linux macOS Windows Cloud Service Web

 Blazor App
Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).
C# Linux macOS Windows Cloud Web

 ASP.NET Web Application (.NET Framework)

SELECT TO CREATE

Next

Updated on May 2020

8/25/2020

Copyright 2020 www.elain-tech.com Syed Awase Khirni

52

Configure your new project

ASP.NET Core Web Application

C#

Linux

macOS

Windows

Cloud

Service

Web

Project name

ex6aspnetcoreangularstack


Location

D:\awase-harddisk\CT\Csharp2017\aspnetcore3x\codeplay\ex6aspnetcoreangularstack\

...

Solution

Create new solution

Solution name 

ex6aspnetcoreangularstack

☒ Place solution and project in the same directory

Back

Create

Create a new ASP.NET Core web application

.NET Core

ASP.NET Core 3.1

Empty
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

Web Application
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

Web Application (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

Angular
A project template for creating an ASP.NET Core application with Angular

React.js

[Get additional project templates](#)

Authentication
No Authentication
[Change](#)

Advanced

☒ Configure for HTTPS

☐ Enable Docker Support
(Requires Docker Desktop)

Linux

Author: Microsoft
Source: Templates 3.1.8

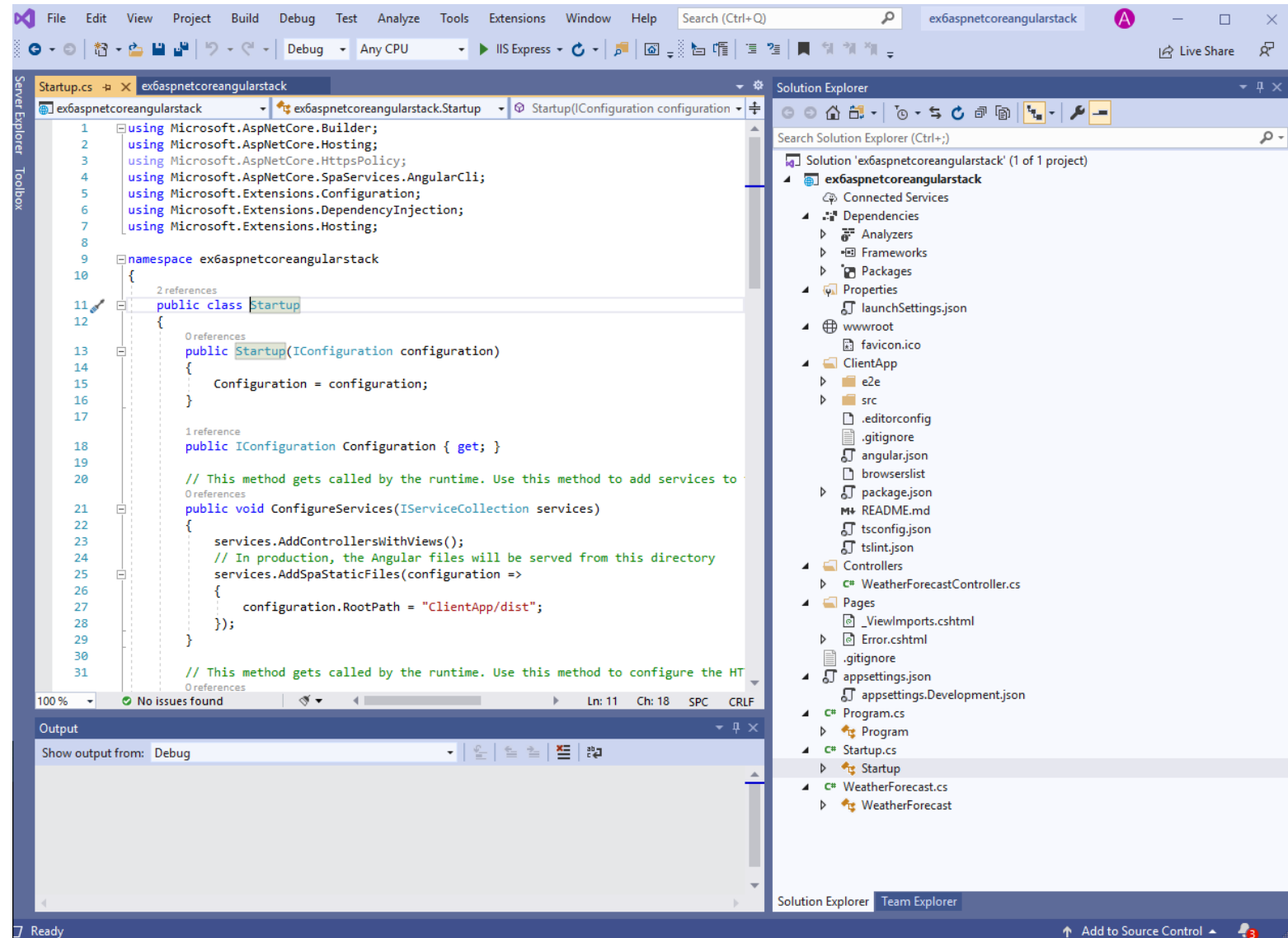
Back

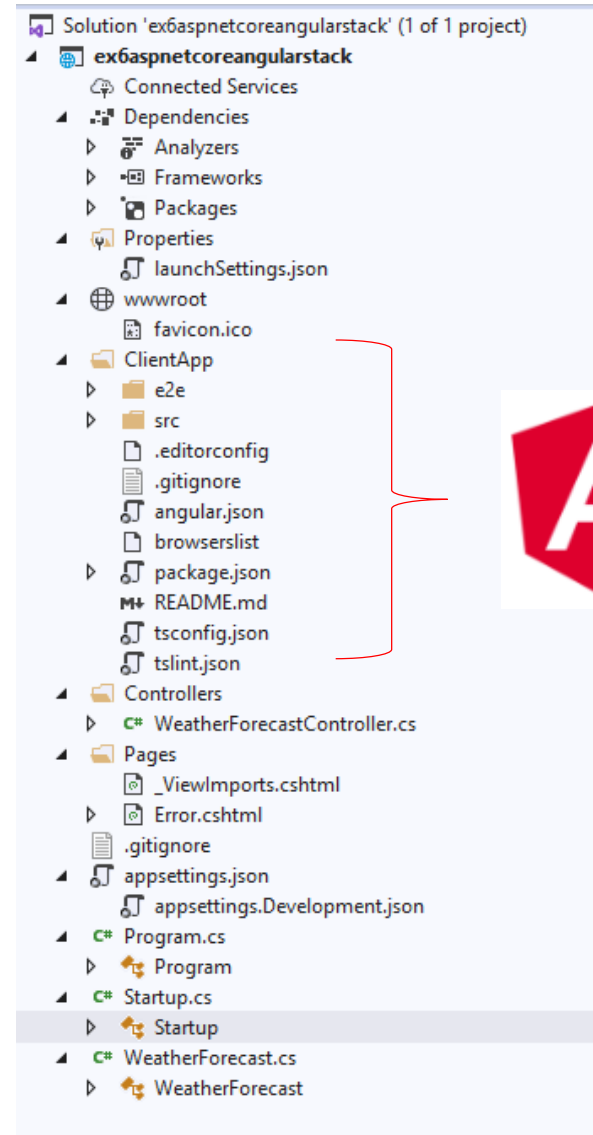
Create

Please read terms of use for authorized access

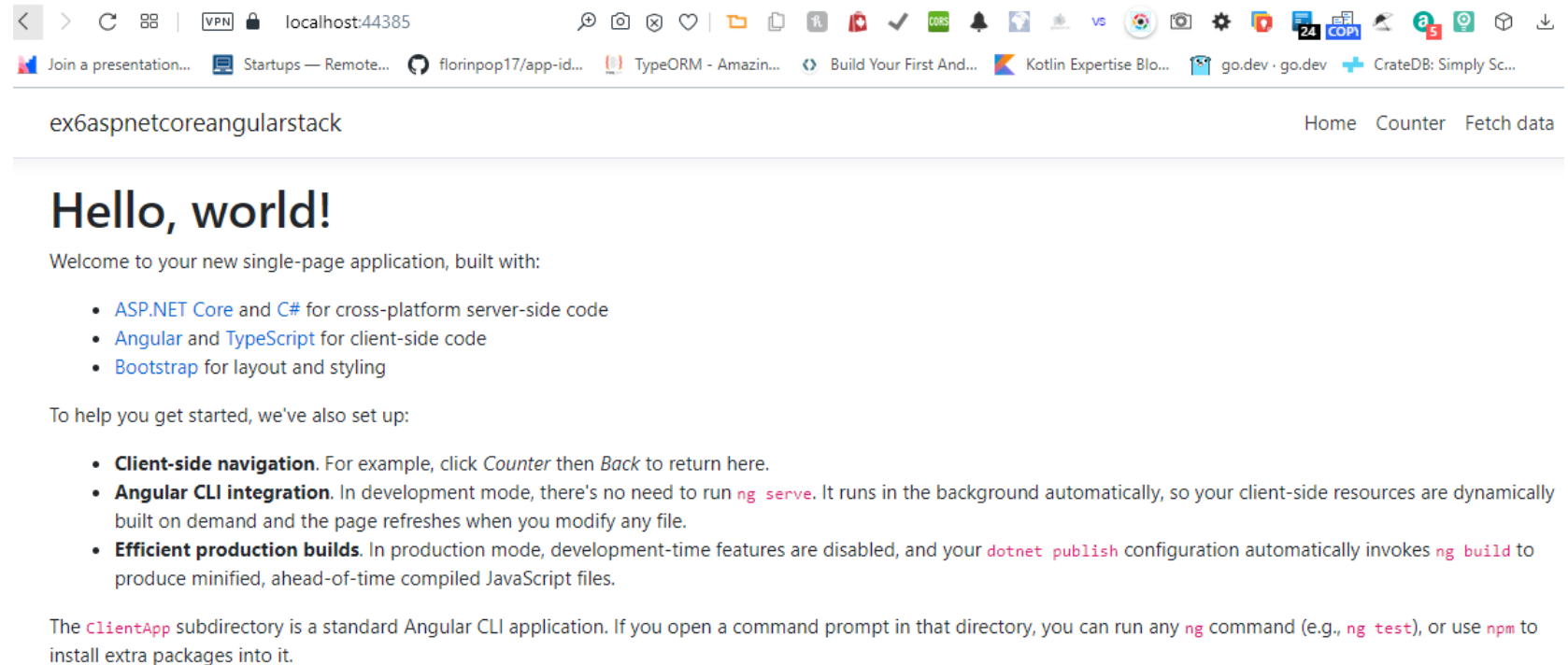
Original Series

Updated on May 2020





http://localhost:44385



The screenshot shows a web browser window with the address bar displaying 'localhost:44385'. The page title is 'ex6aspnetcoreangularstack'. The main content area displays 'Hello, world!' in a large, bold font. Below this, a welcome message states 'Welcome to your new single-page application, built with:' followed by a bulleted list of technologies: ASP.NET Core, C#, Angular, TypeScript, and Bootstrap. Further down, another section titled 'To help you get started, we've also set up:' contains three bullet points: 'Client-side navigation', 'Angular CLI integration', and 'Efficient production builds'. The bottom of the page mentions the 'ClientApp' subdirectory and provides instructions on running 'ng' commands or using 'npm'.

ex6aspnetcoreangularstack

Home Counter Fetch data

Hello, world!

Welcome to your new single-page application, built with:

- [ASP.NET Core](#) and [C#](#) for cross-platform server-side code
- [Angular](#) and [TypeScript](#) for client-side code
- [Bootstrap](#) for layout and styling

To help you get started, we've also set up:

- **Client-side navigation.** For example, click *Counter* then *Back* to return here.
- **Angular CLI integration.** In development mode, there's no need to run `ng serve`. It runs in the background automatically, so your client-side resources are dynamically built on demand and the page refreshes when you modify any file.
- **Efficient production builds.** In production mode, development-time features are disabled, and your `dotnet publish` configuration automatically invokes `ng build` to produce minified, ahead-of-time compiled JavaScript files.

The `ClientApp` subdirectory is a standard Angular CLI application. If you open a command prompt in that directory, you can run any `ng` command (e.g., `ng test`), or use `npm` to install extra packages into it.

http://localhost:44385/counter

ex6aspnecoreangularstackHome Counter Fetch data

Counter

This is a simple example of an Angular component.

Current count: 0

Increment

http://localhost:44385/fetch-data

localhost:44385/fetch-data

Join a presentation... Startups — Remote... florinpop17/app-id... TypeORM - Amazin... Build Your First And... Kotlin Expertise Blo... go.dev · go.dev + CrateDB: Simply Sc...

ex6aspnetcoreangularstack

Home Counter Fetch data

Weather forecast

This component demonstrates fetching data from the server.

Date	Temp. (C)	Temp. (F)	Summary
2020-08-26T12:47:29.285781-04:00	39	102	Scorching
2020-08-27T12:47:29.2870991-04:00	54	129	Freezing
2020-08-28T12:47:29.2871119-04:00	18	64	Chilly
2020-08-29T12:47:29.2871136-04:00	19	66	Mild
2020-08-30T12:47:29.2871144-04:00	30	85	Sweltering

Please read terms of use for authorized access

Original Series

Updated on May 2020

SYED AWASE KHIRNI



Example 7: ASP.NET Core React Stack Application

Learning Outcomes:

1. Understand the steps to create ASP.NET Core 3.x react stack application
2. Understand the web application structure
3. Understand the key dependencies for ASP.NET Core 3.x react stack application
4. Executing ASP.NET Core 3.x react stack application and rendering it on <http://localhost:44360>

Create a new project

Recent project templates

-  ASP.NET Core Web Application C#
-  Console App (.NET Core) C#

SELECT TO CREATE

Search for templates (Alt+S)

All languages

All platforms

All project types



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

C# Linux macOS Windows Console



Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

Visual Basic Linux macOS Windows Console



ASP.NET Core Web Application

Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

C# Linux macOS Windows Cloud Service Web



Blazor App

Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).

C# Linux macOS Windows Cloud Web



ASP.NET Web Application (.NET Framework)

Next

Configure your new project

ASP.NET Core Web Application

C#

Linux

macOS

Windows

Cloud

Service

Web

Project name

ex7aspnetcorereactstackprj

Location

D:\awase-harddisk\CT\Csharp2017\aspnetcore3x\codeplay\ex7aspnetcorereactstack\

...

Solution

Create new solution

Solution name ⓘ

ex7aspnetcorereactstackprj

☒ Place solution and project in the same directory


Back


Create


Create a new ASP.NET Core web application


.NET Core


ASP.NET Core 3.1


**API**
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

**Web Application**
A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

**Web Application (Model-View-Controller)**
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

**Angular**
A project template for creating an ASP.NET Core application with Angular

**React.js**
A project template for creating an ASP.NET Core application with React.js

**React.js and Redux**
A project template for creating an ASP.NET Core application with React.js and Redux

Authentication
No Authentication
[Change](#)

Advanced
☒ Configure for HTTPS
☐ Enable Docker Support
(Requires Docker Desktop)

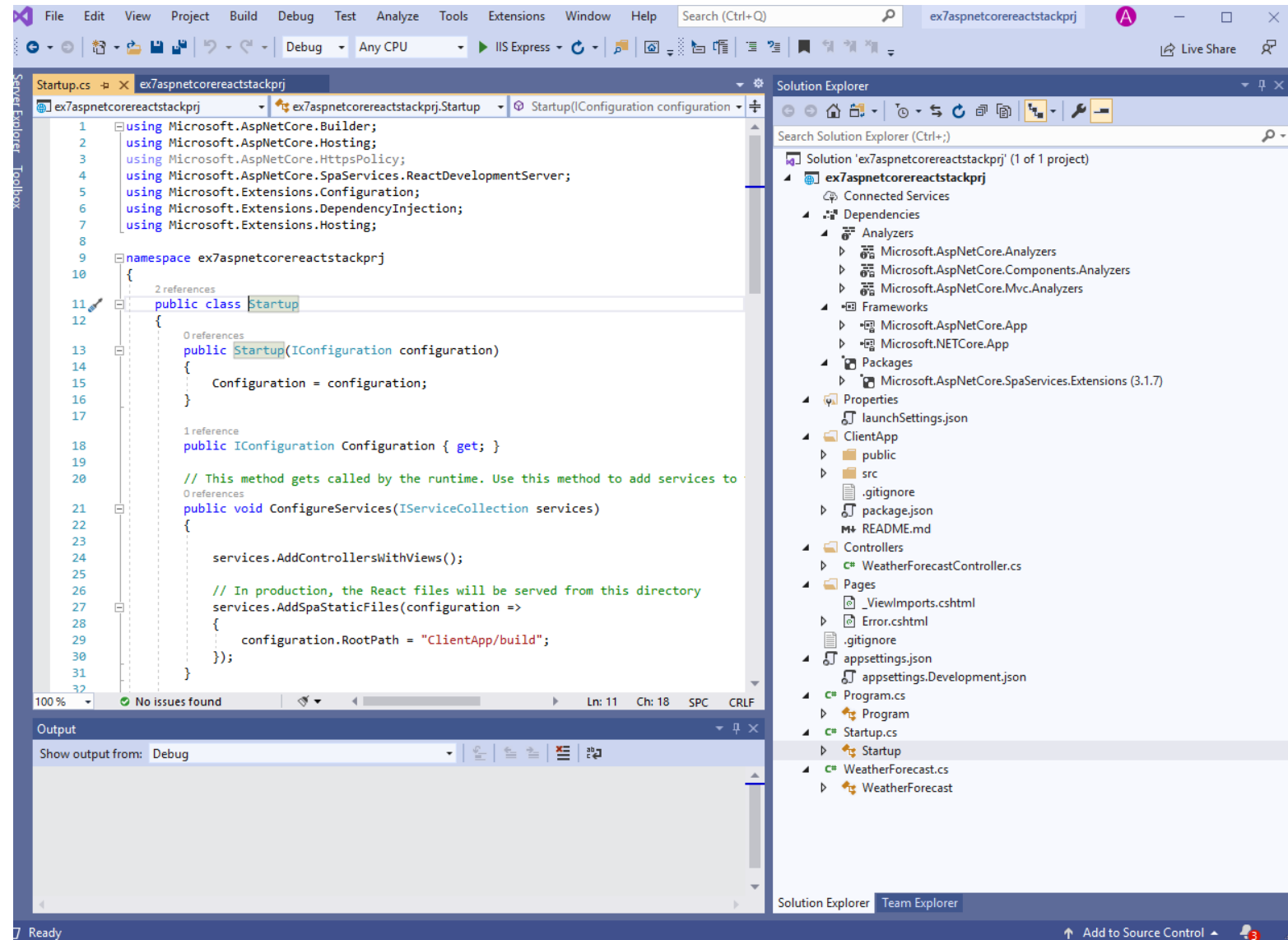
Linux

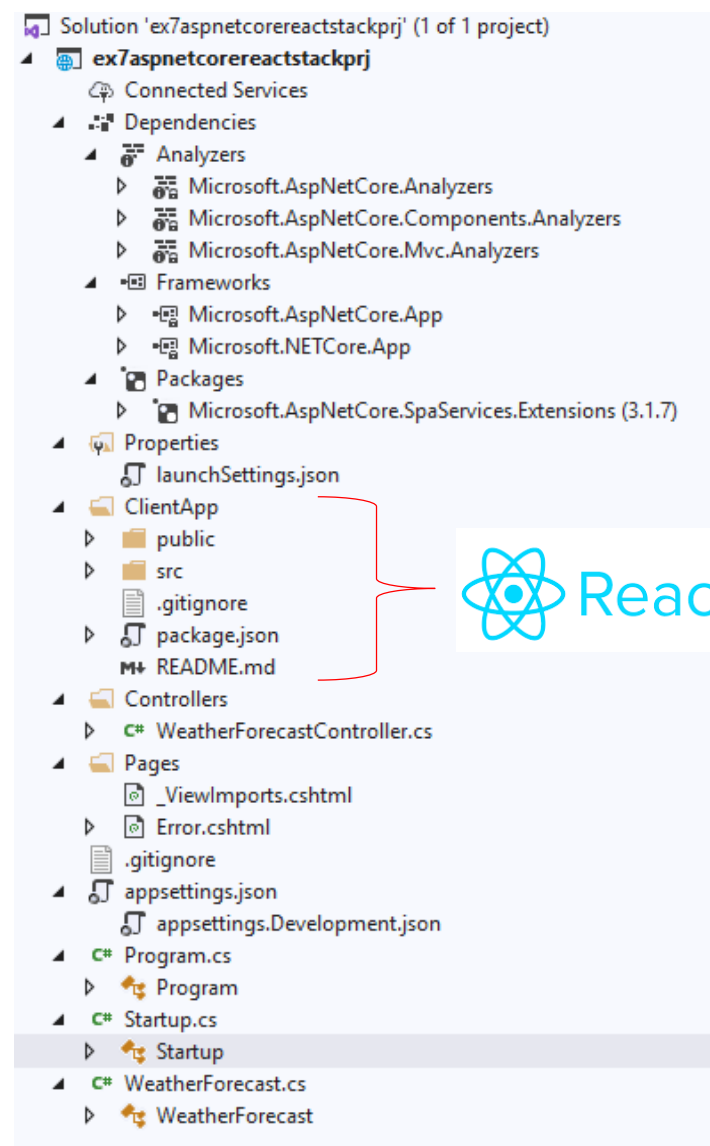
Author: Microsoft
Source: Templates 3.1.8

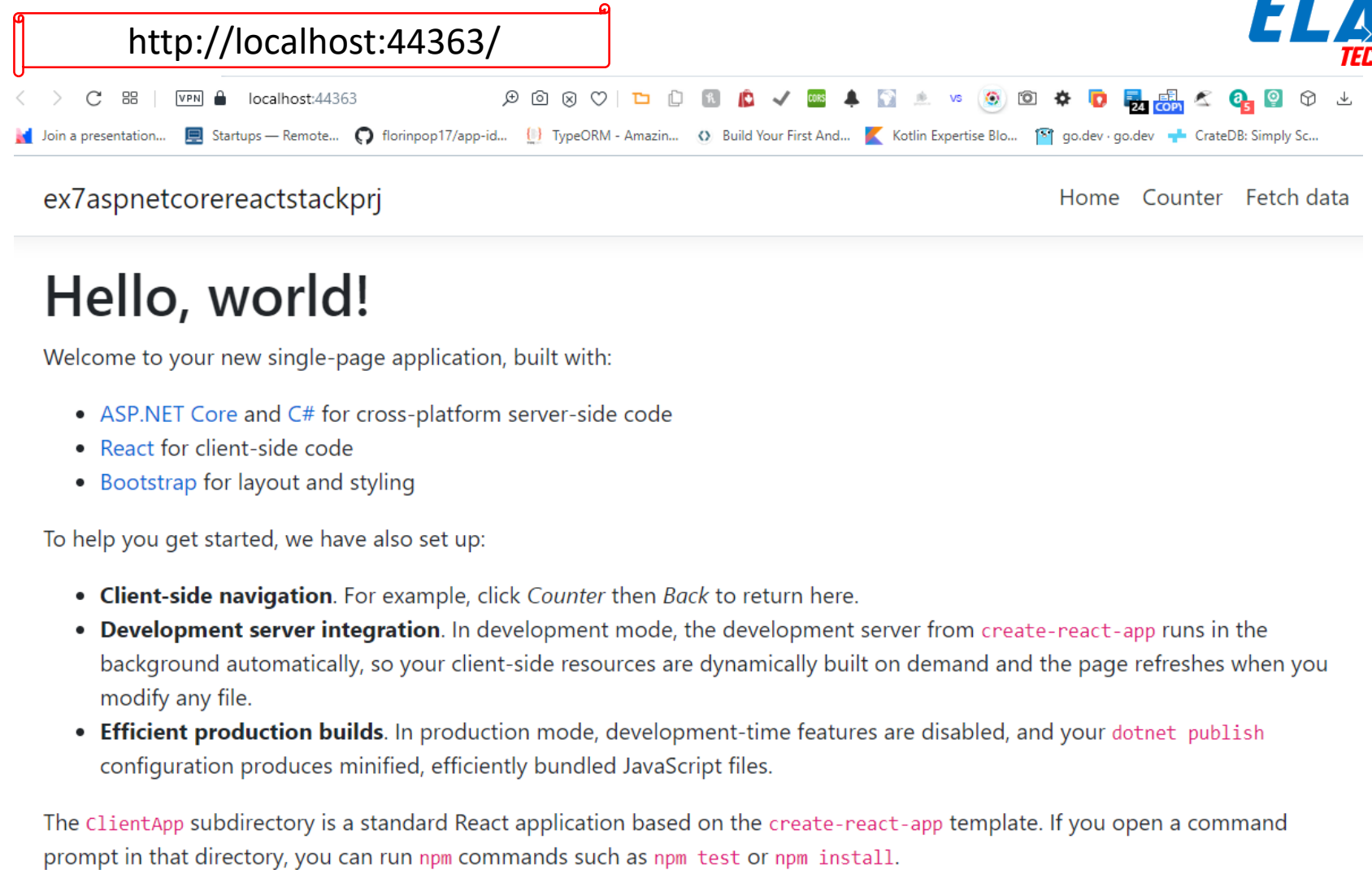
Get additional project templates

Back

Create







<http://localhost:44363/>

ex7aspnetcorereactstackprj [Home](#) [Counter](#) [Fetch data](#)

Hello, world!

Welcome to your new single-page application, built with:

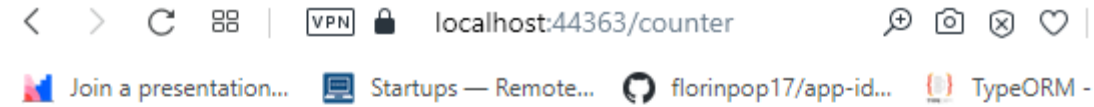
- [ASP.NET Core](#) and [C#](#) for cross-platform server-side code
- [React](#) for client-side code
- [Bootstrap](#) for layout and styling

To help you get started, we have also set up:

- **Client-side navigation.** For example, click *Counter* then *Back* to return here.
- **Development server integration.** In development mode, the development server from [create-react-app](#) runs in the background automatically, so your client-side resources are dynamically built on demand and the page refreshes when you modify any file.
- **Efficient production builds.** In production mode, development-time features are disabled, and your [dotnet publish](#) configuration produces minified, efficiently bundled JavaScript files.

The [ClientApp](#) subdirectory is a standard React application based on the [create-react-app](#) template. If you open a command prompt in that directory, you can run [npm](#) commands such as [npm test](#) or [npm install](#).

http://localhost:44363/counter



ex7aspnetcorereactstackprj

Counter

This is a simple example of a React component.

Current count: 3

Increment