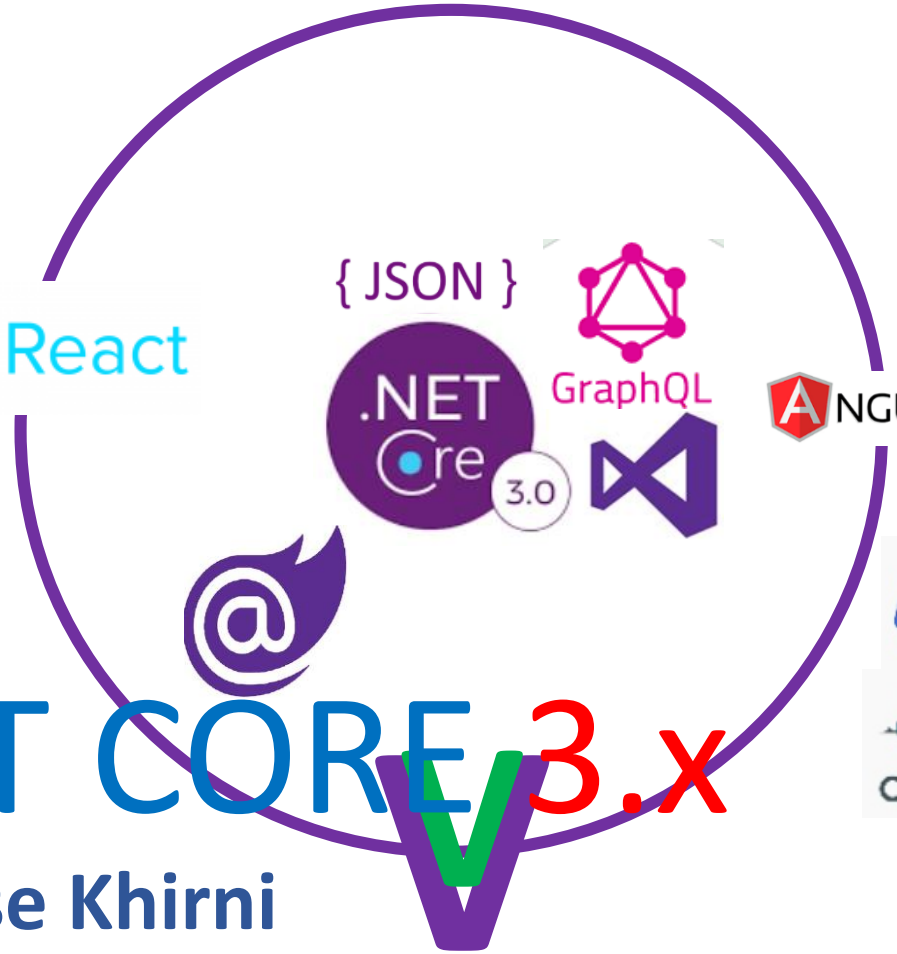


Ground Up Series

ASP.NET CORE V3.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				

Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

.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)

Original Series

Updated on May 2020 https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

.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

Original Series

Updated on May 2020 https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

.NET Core 3.x

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

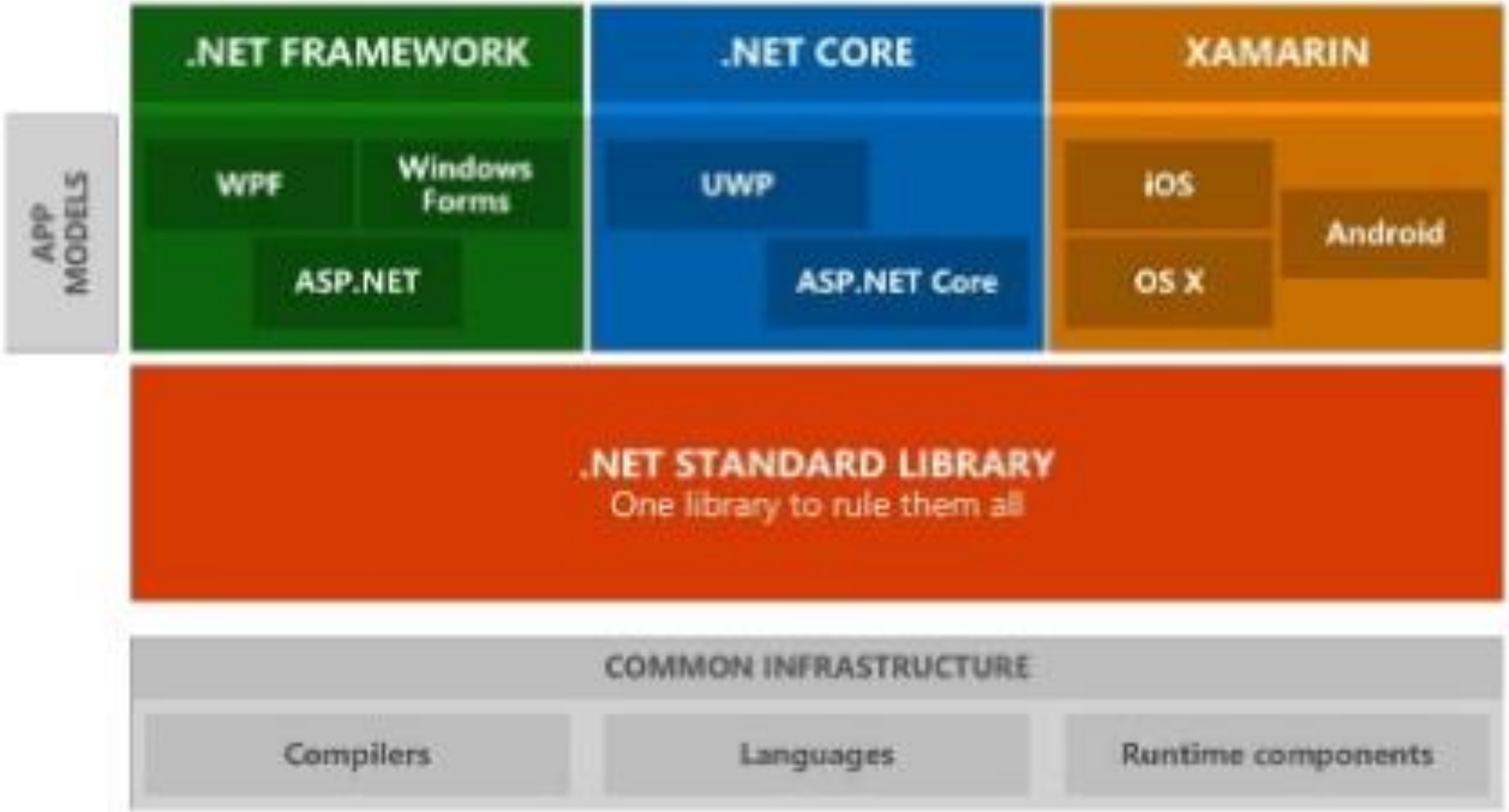
ASP.NET	API, MVC, Razor Pages, Blazor Server, Blazor client, gRPC, SignalR
Windows Forms	
WPF	
Entity Framework	.Net Standard 2.1, x-plat lightweight, C#8 (Async streams, nullable reference types) CosmosDB
WCF	Core WCF, Rebuild to gRPC

Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

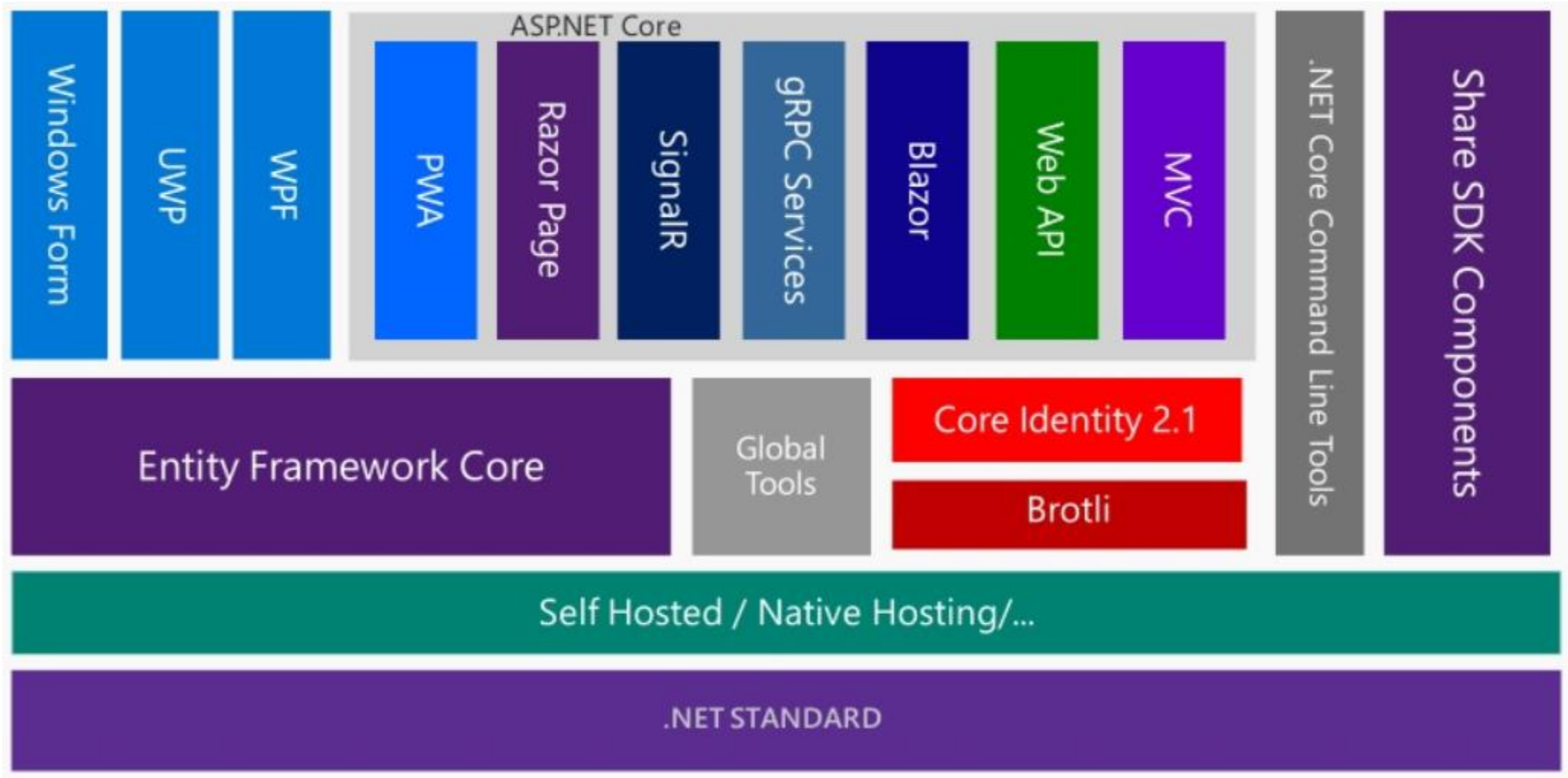
Current Platform Requirements

- High Performance Framework/Platform, Highly Secure and Easily Scalable platform.
- Efficient memory management.
- Cross-platform support with CICD – 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.



Original Series

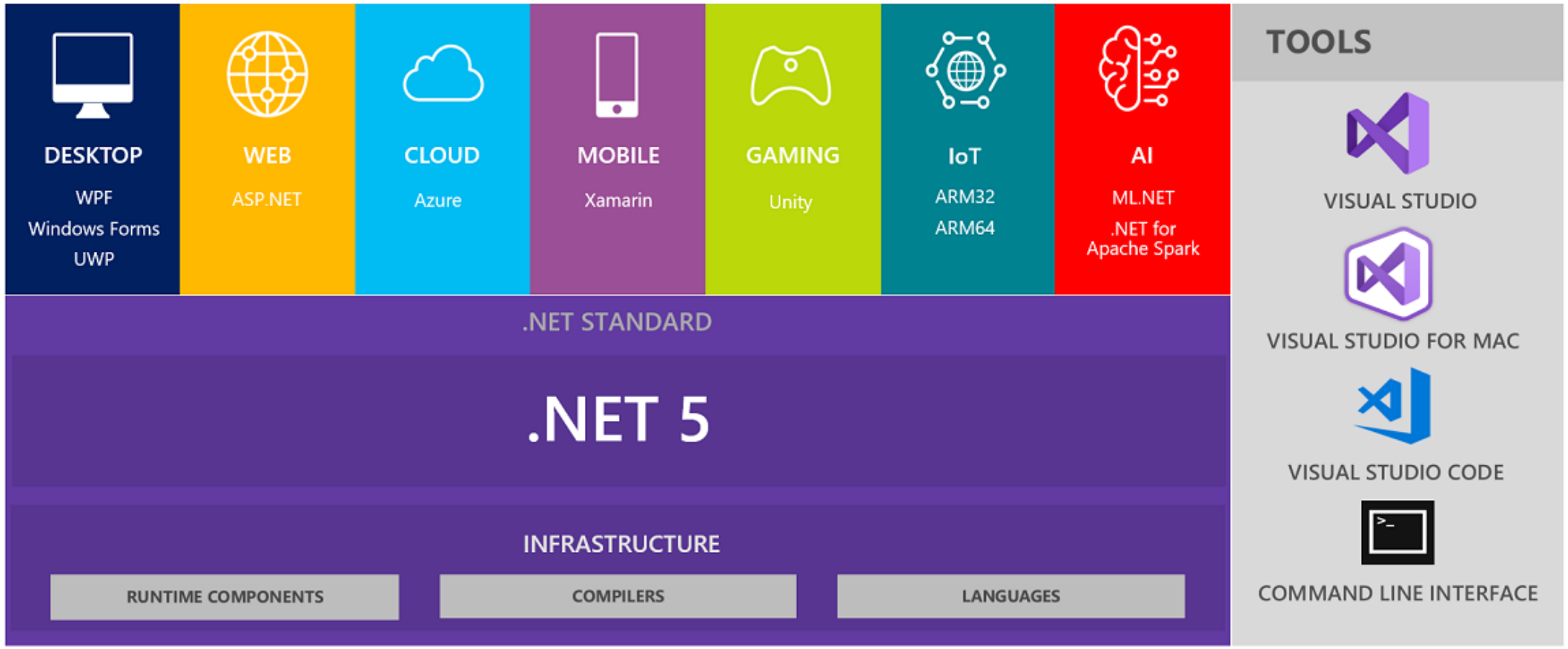
Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x



(source: Microsoft)

.NET – A unified platform

ASP.NET vNext → ASP.NET MVC6 → ASP.NET 5 → ASP.NET Core 1.0 → ASP.NET Core 1.1 → ASP.NET Core 2.0 → ASP.NET Core 2.1 → ASP.NET Core 2.2 → ASP.NET Core 3.0 → ASP.NET Core 3.1 → .NET 5



(source: Microsoft)

Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

.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)

Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

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

Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

.NET 5

.NET

.NET5

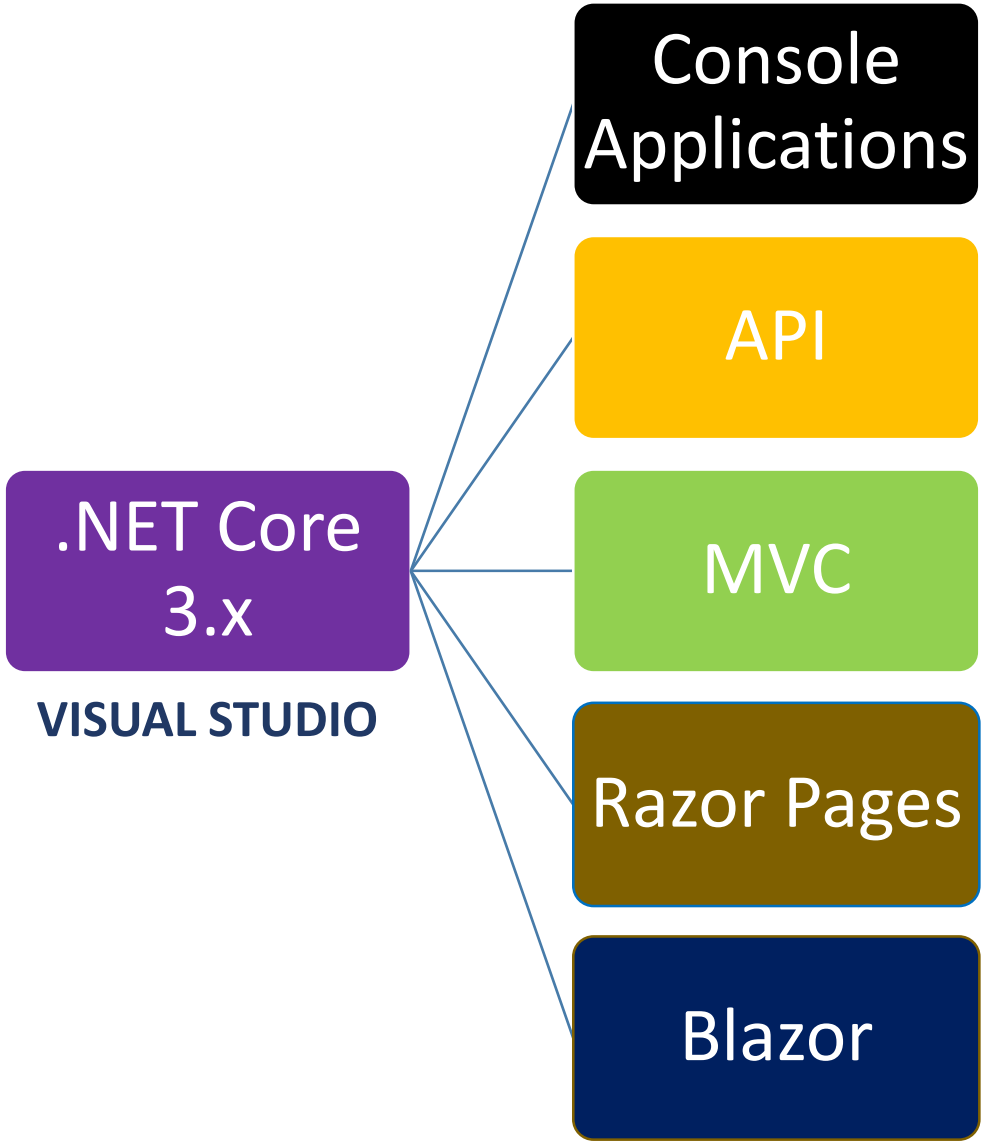
ASP.NET CORE 3.0

- .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 webscoket 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

Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

Original Series



PROJECT TEMPLATES

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

Setting up ASP.NET Core 3.x

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

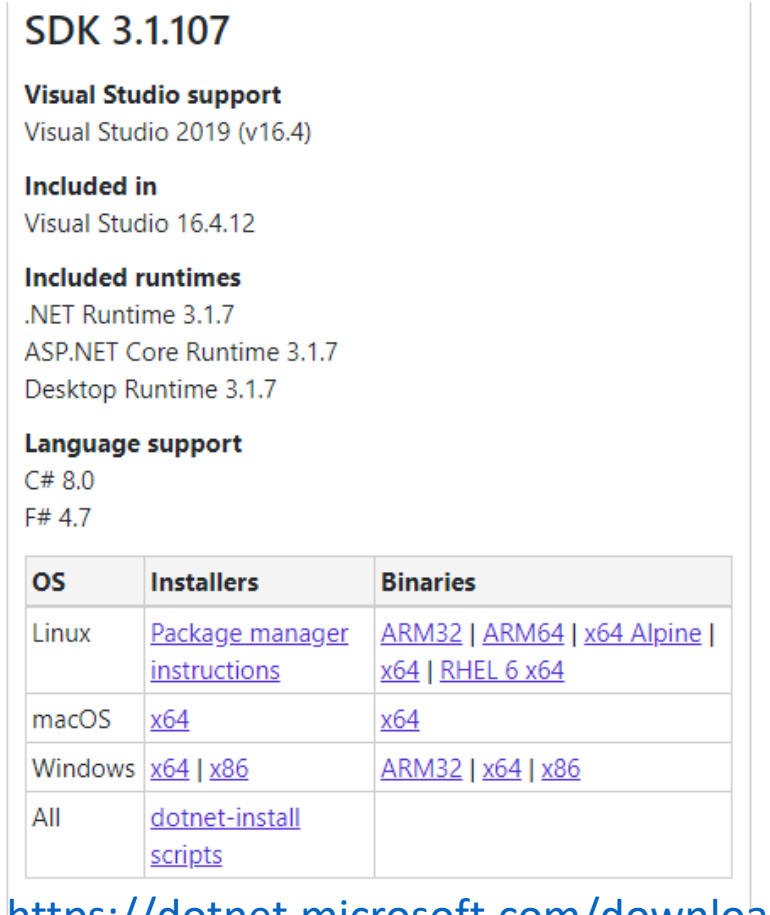
SYED AWASE KHIRNI

Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

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

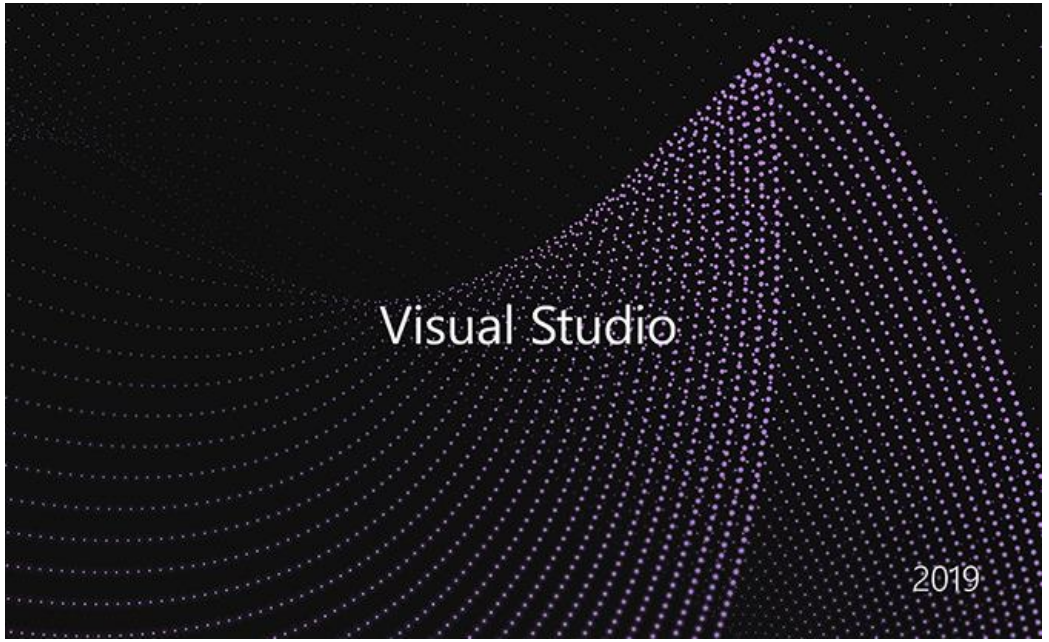
Language support
C# 8.0
F# 4.7

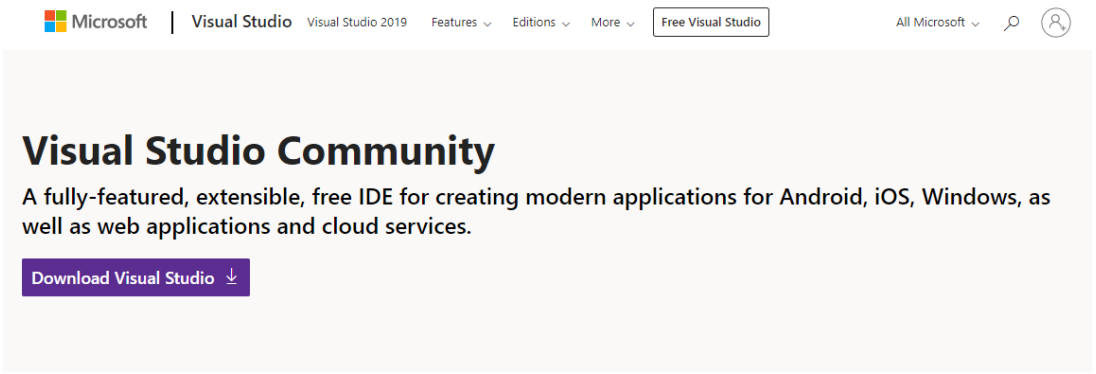
OS	Installers	Binaries
Linux	Package manager instructions	ARM32 | ARM64 | x64 Alpine | x64 | RHEL 6 x64
macOS	x64	x64
Windows	x64 | x86	ARM32 | x64 | x86
All	dotnet-install scripts	

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

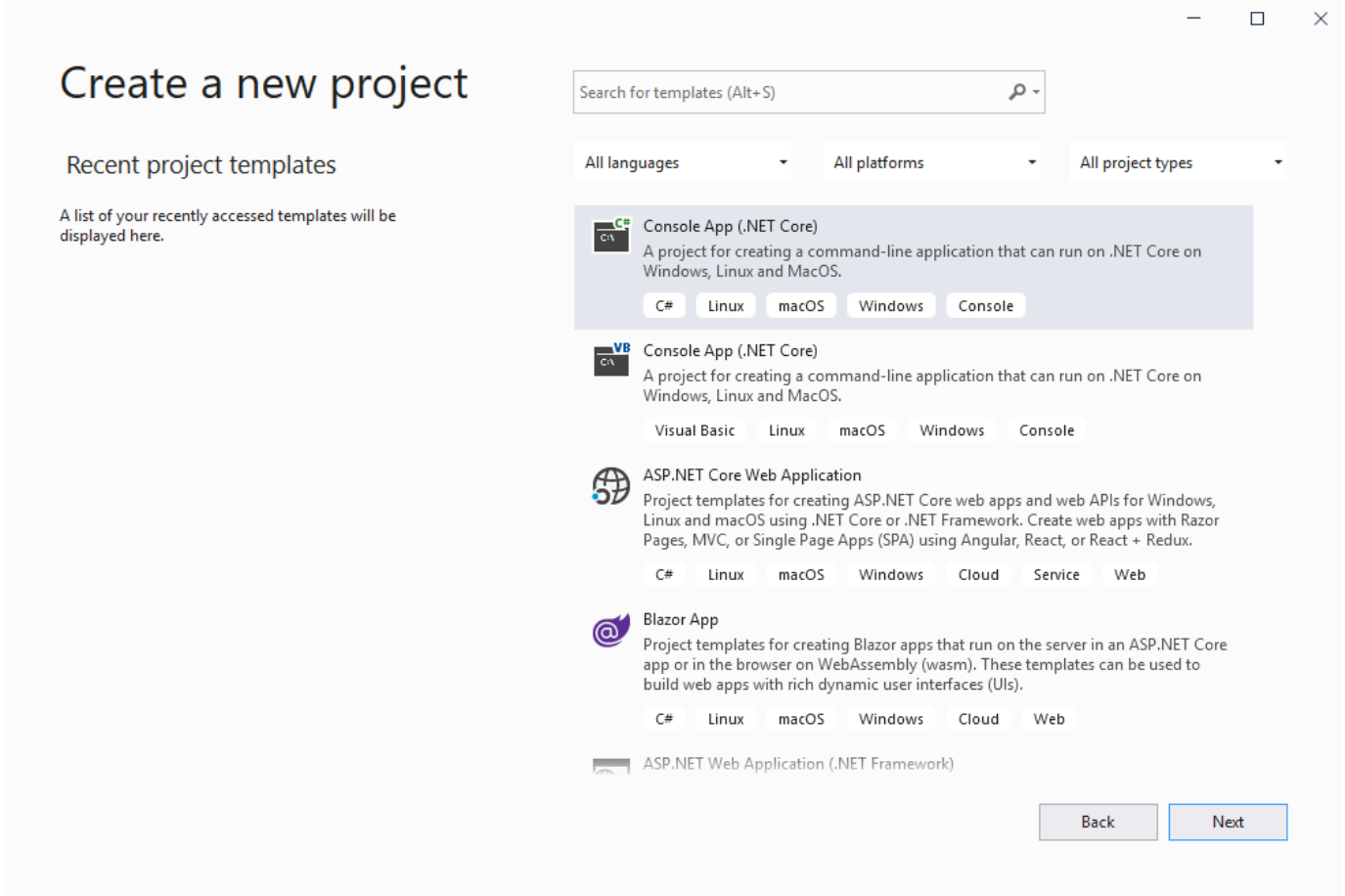
Visual Studio Community IDE 2019

- 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



Original Series

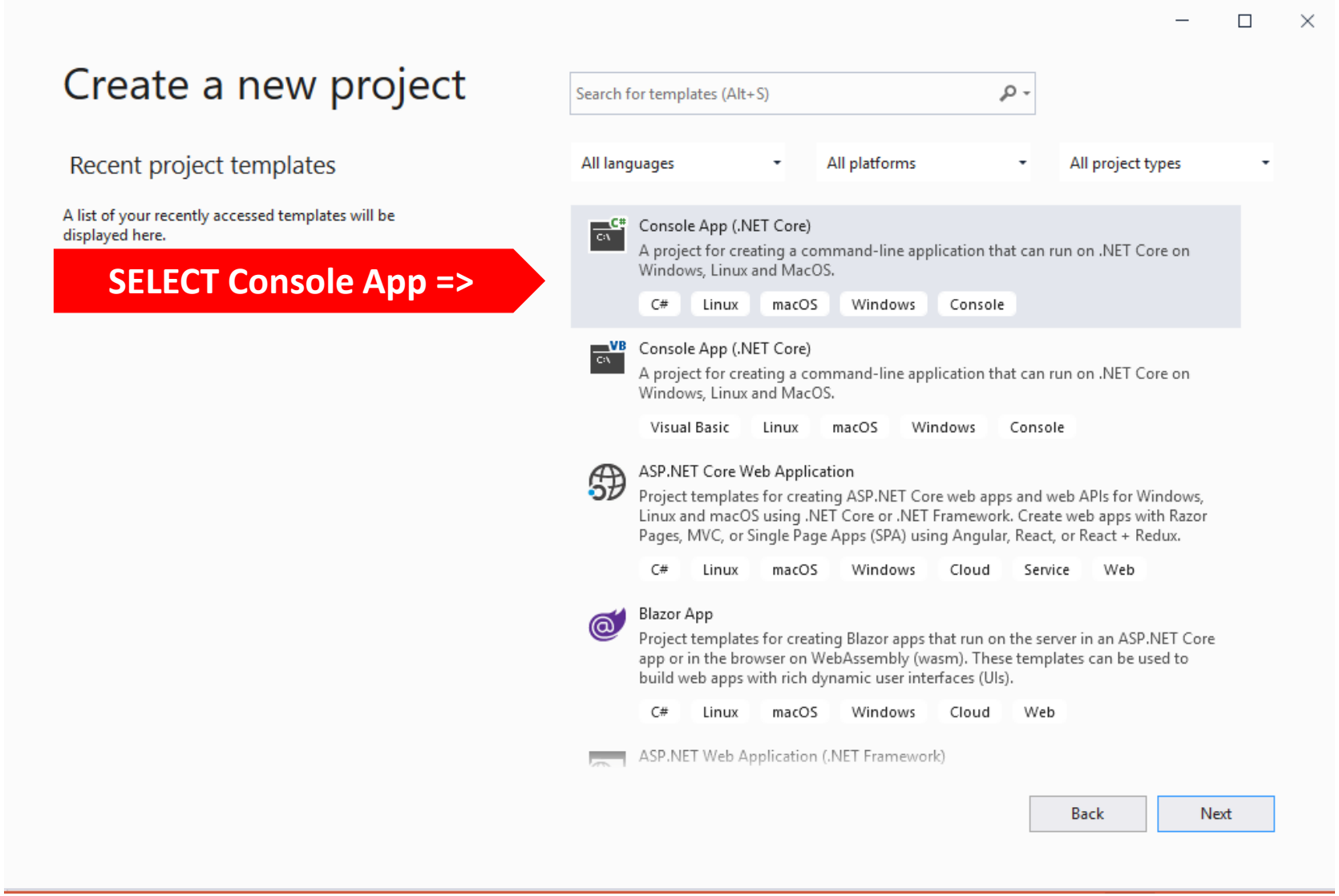
SYED AWASE KHIRNI

Example 1: ConsoleApp .NET CORE

Learning Outcomes:

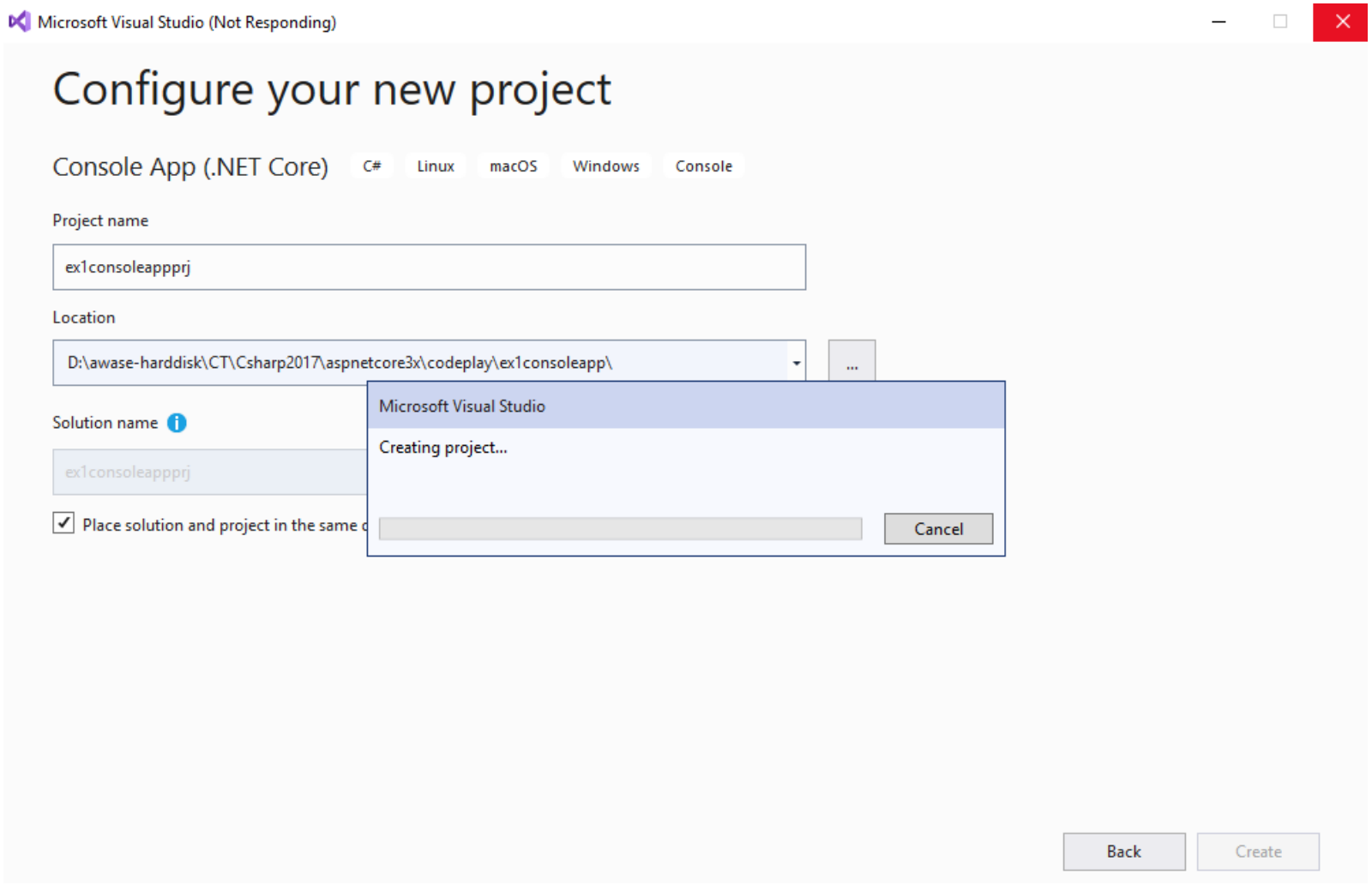
1. **Creating a .Net Core 3.x console application using visual studio 2019.**

Example: 1



Original Series

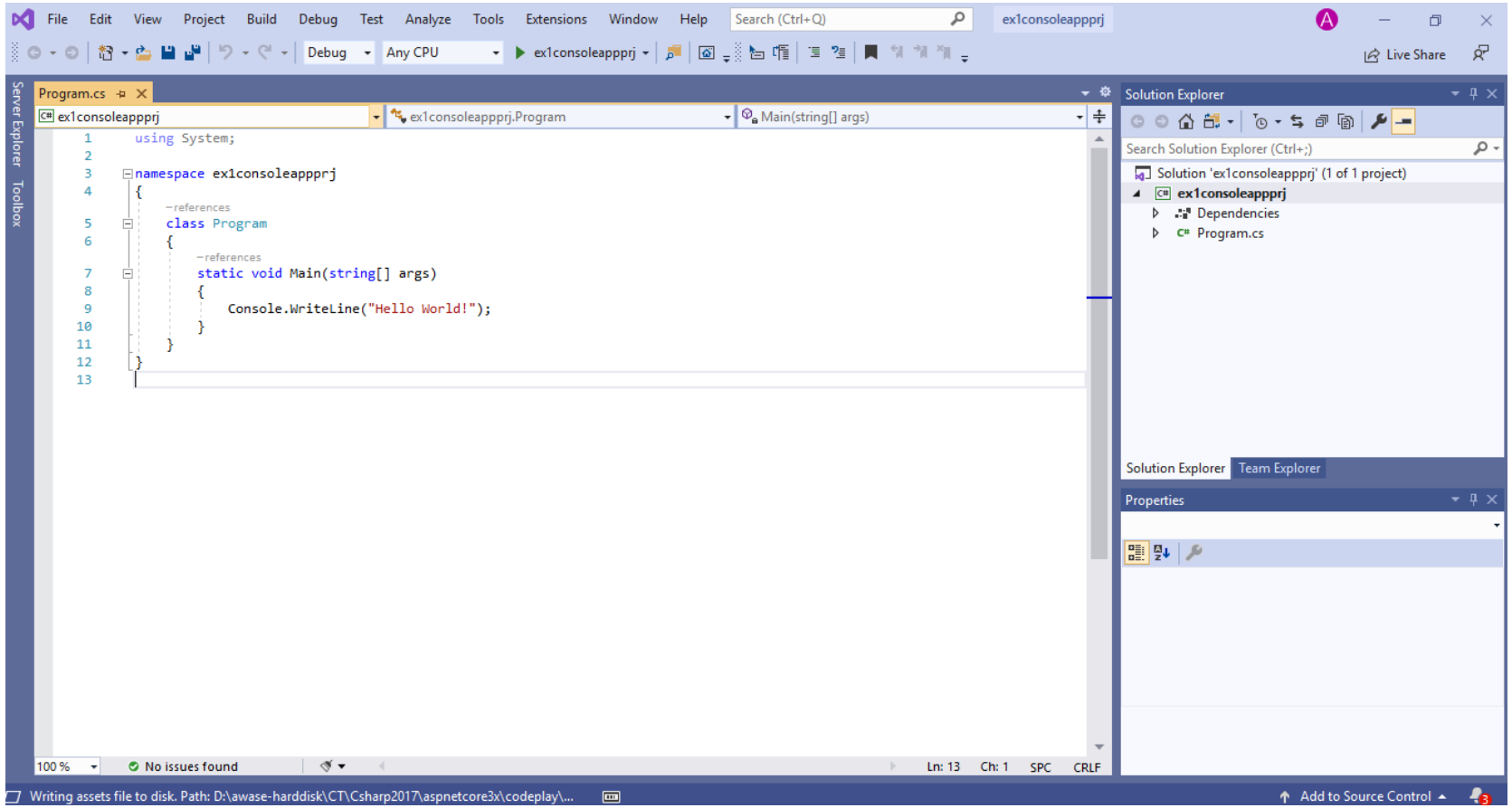
Example: 1



Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

Example: 1

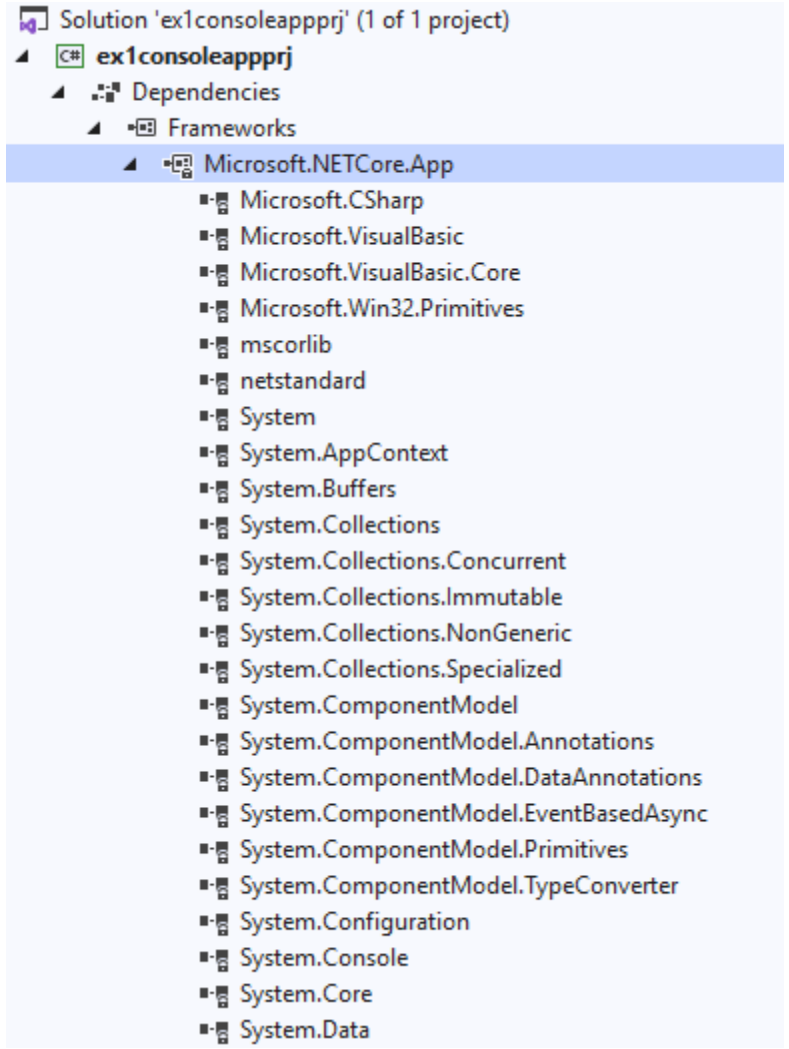


Original Series

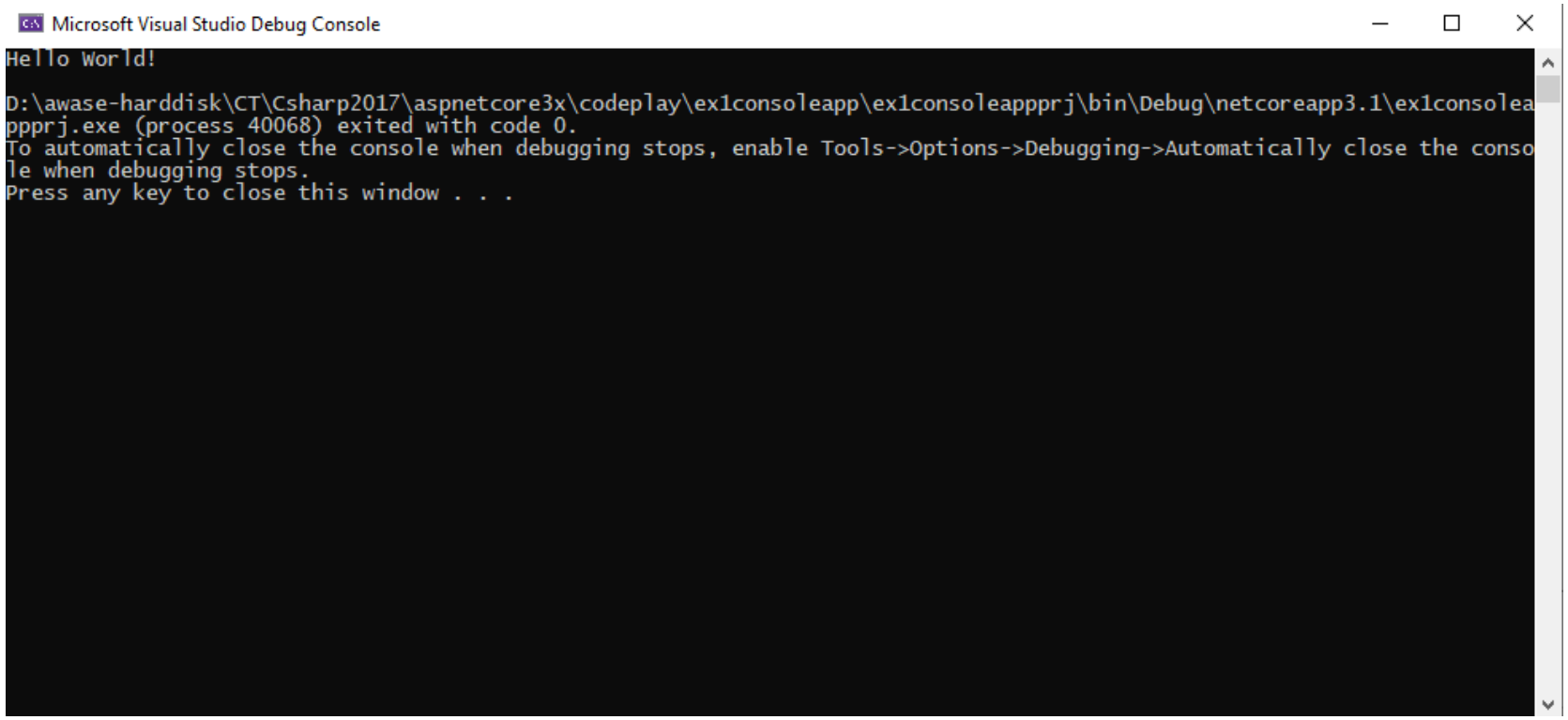
Updated on May 2020

https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

Example:



Example:



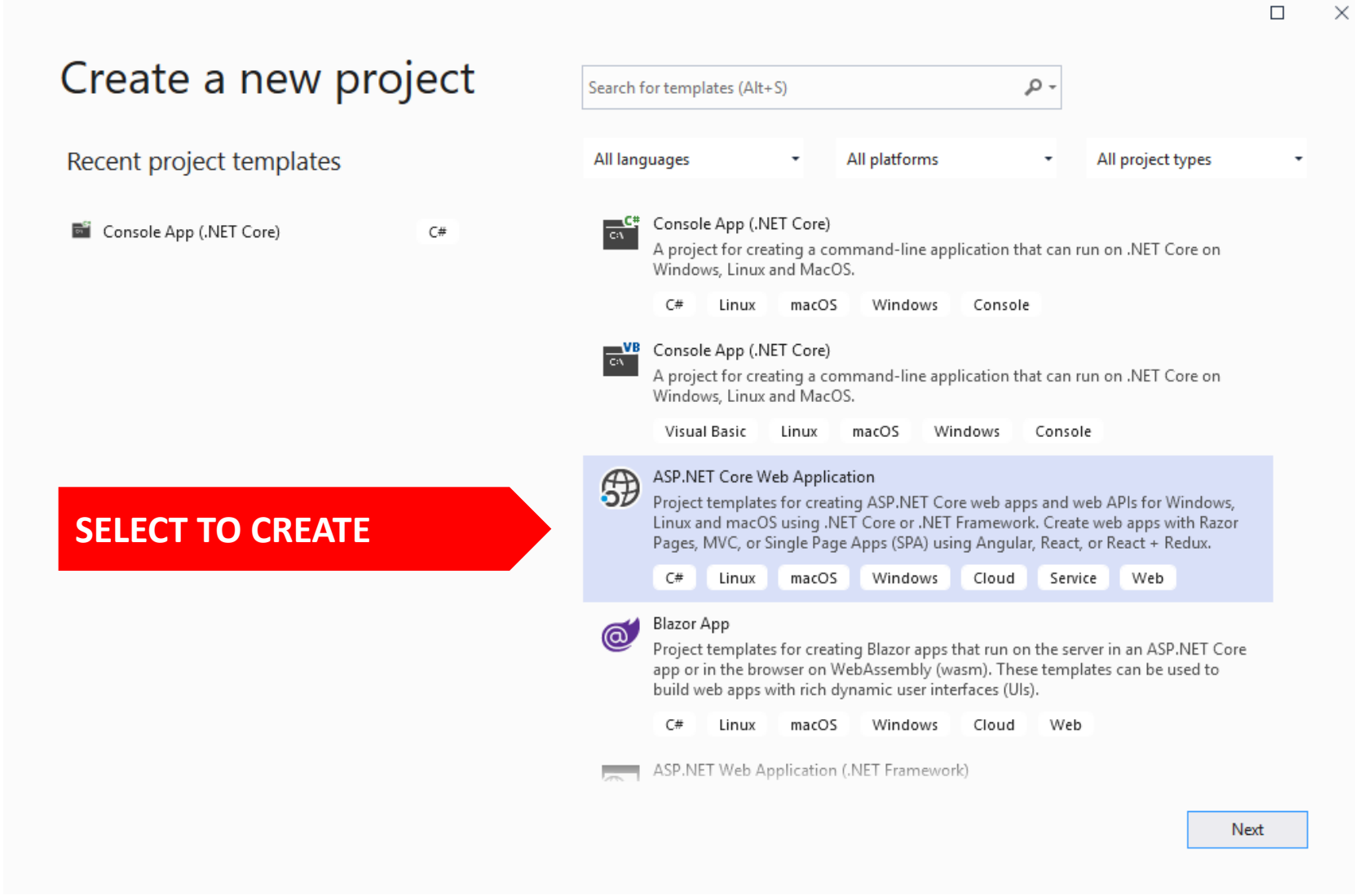
SYED AWASE KHIRNI

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**

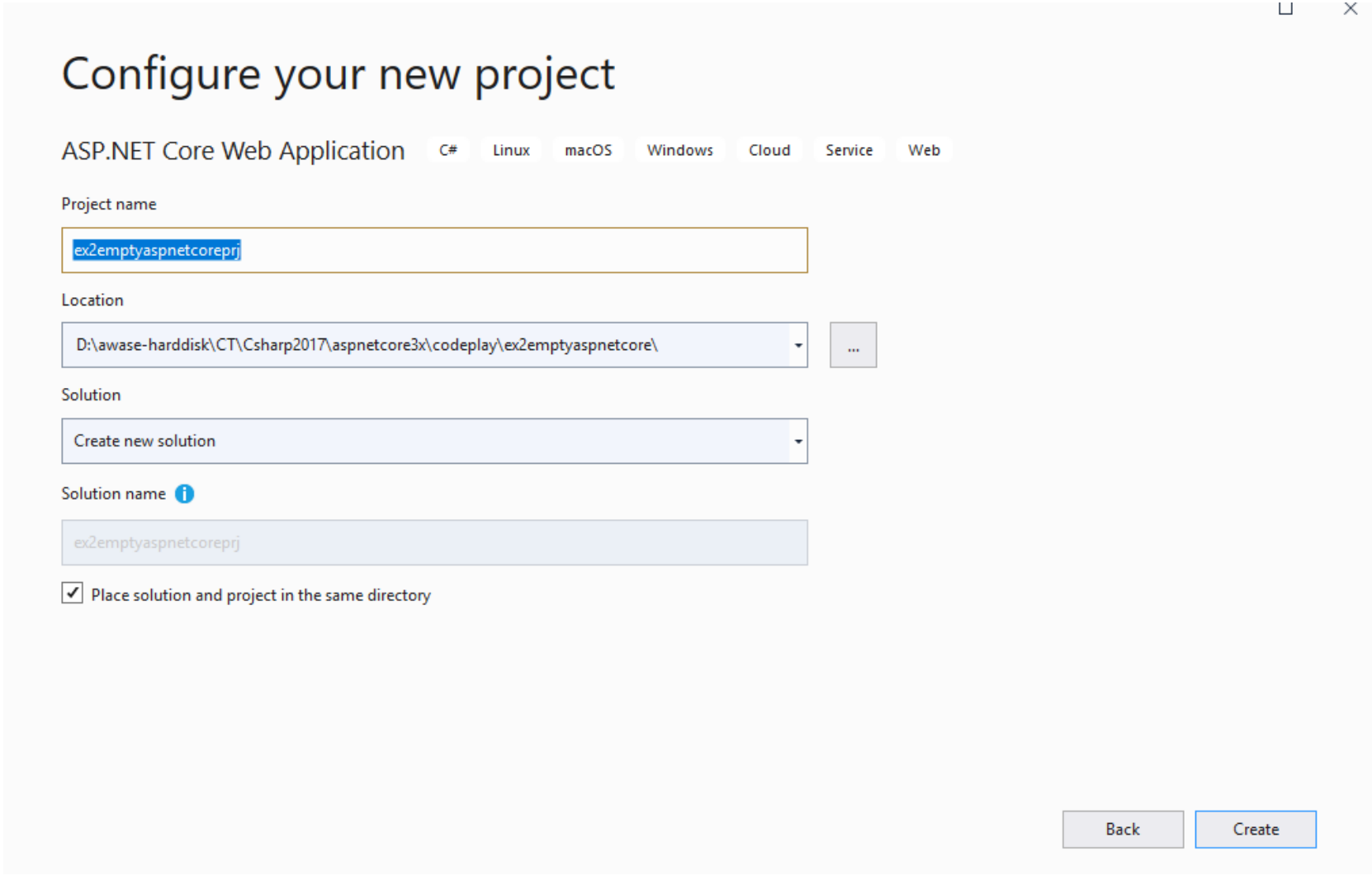
Example: 2



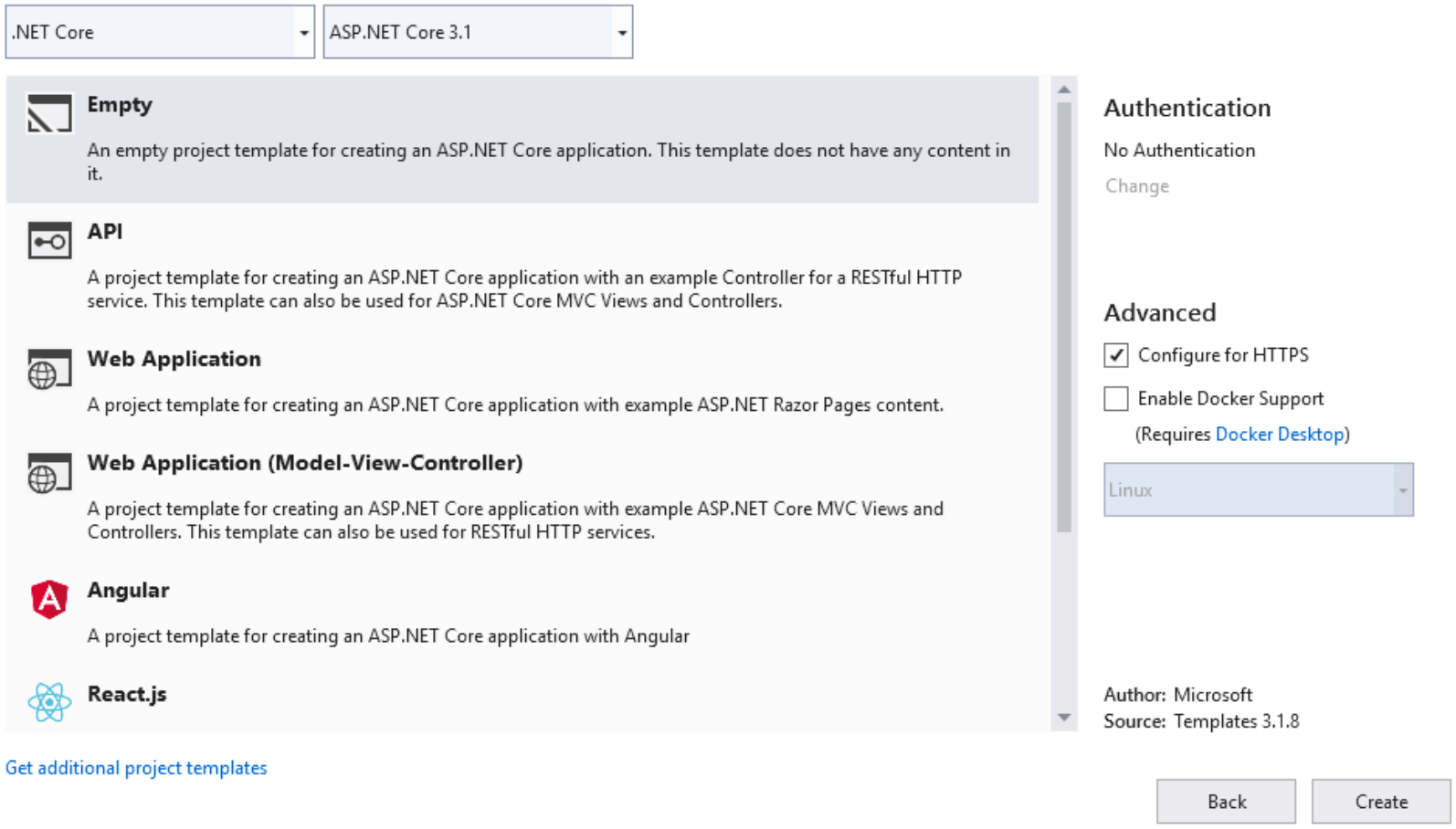
SELECT TO CREATE

Original Series

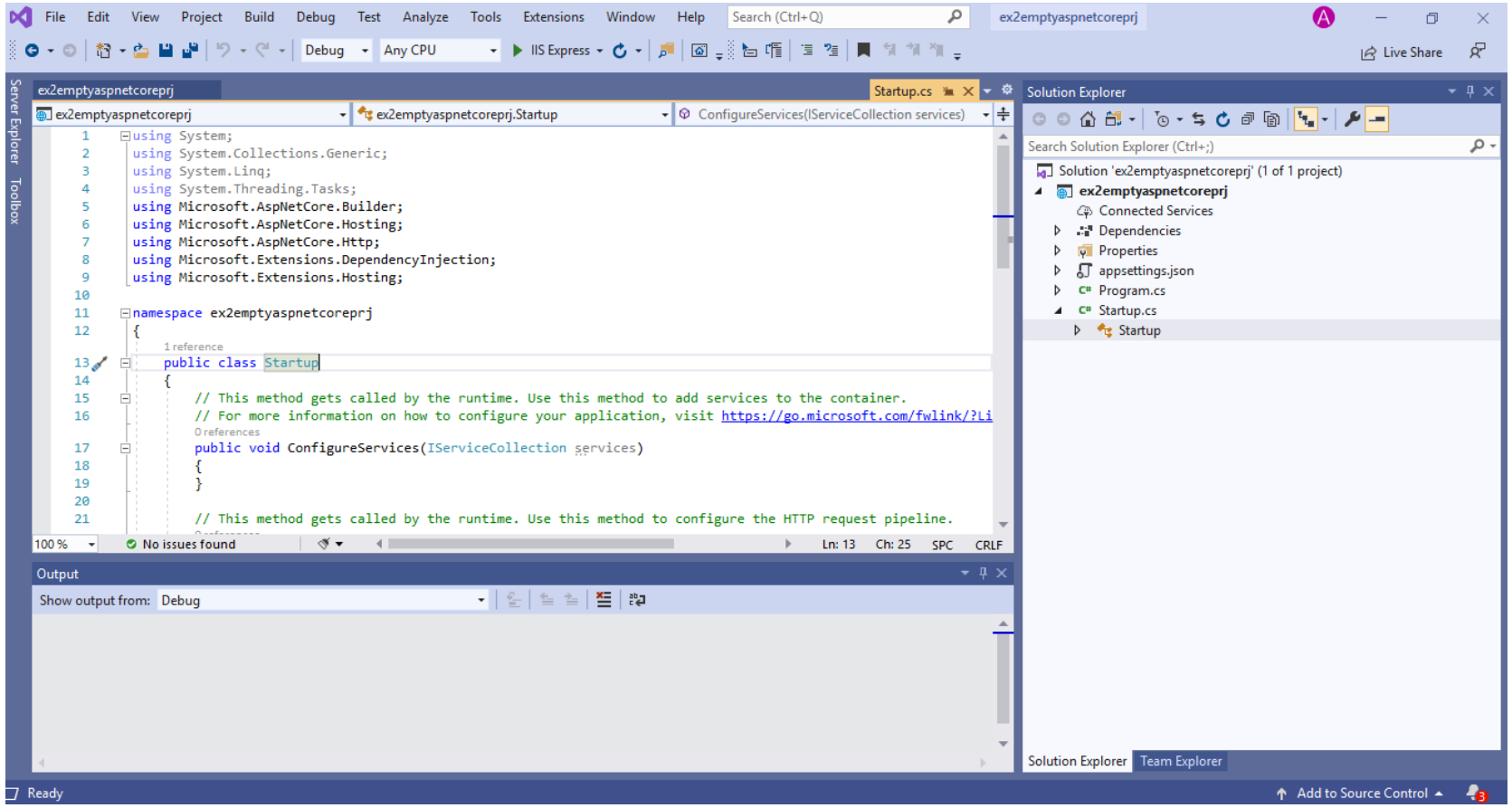
Example: 2



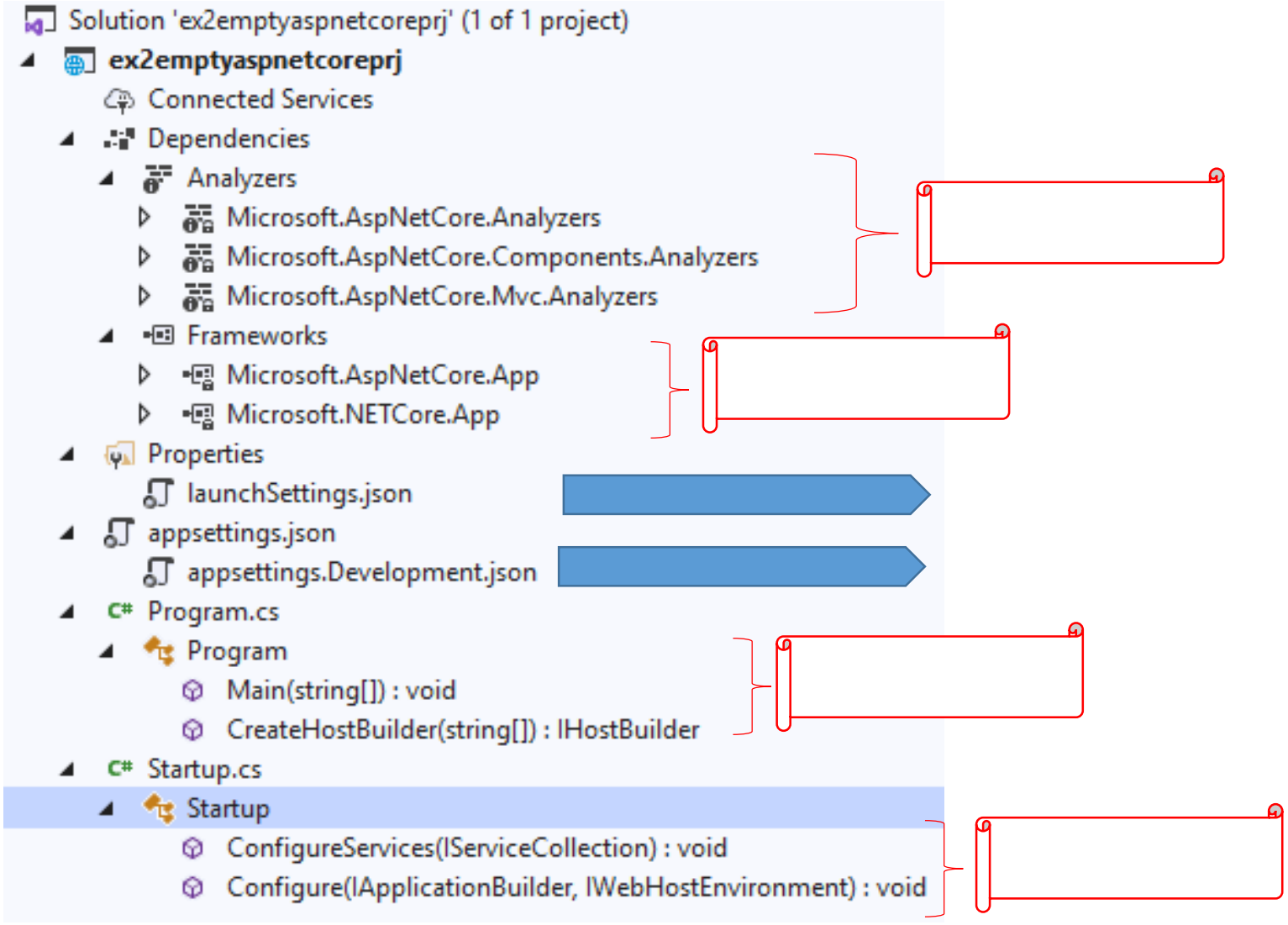
Create a new ASP.NET Core web application



Example: 2

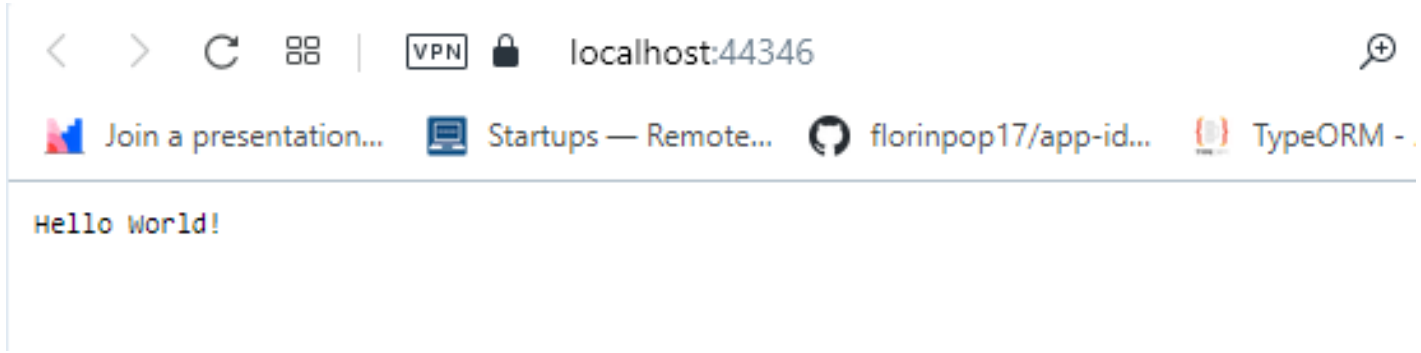


Example: 2



Example: 2

http://localhost:44346



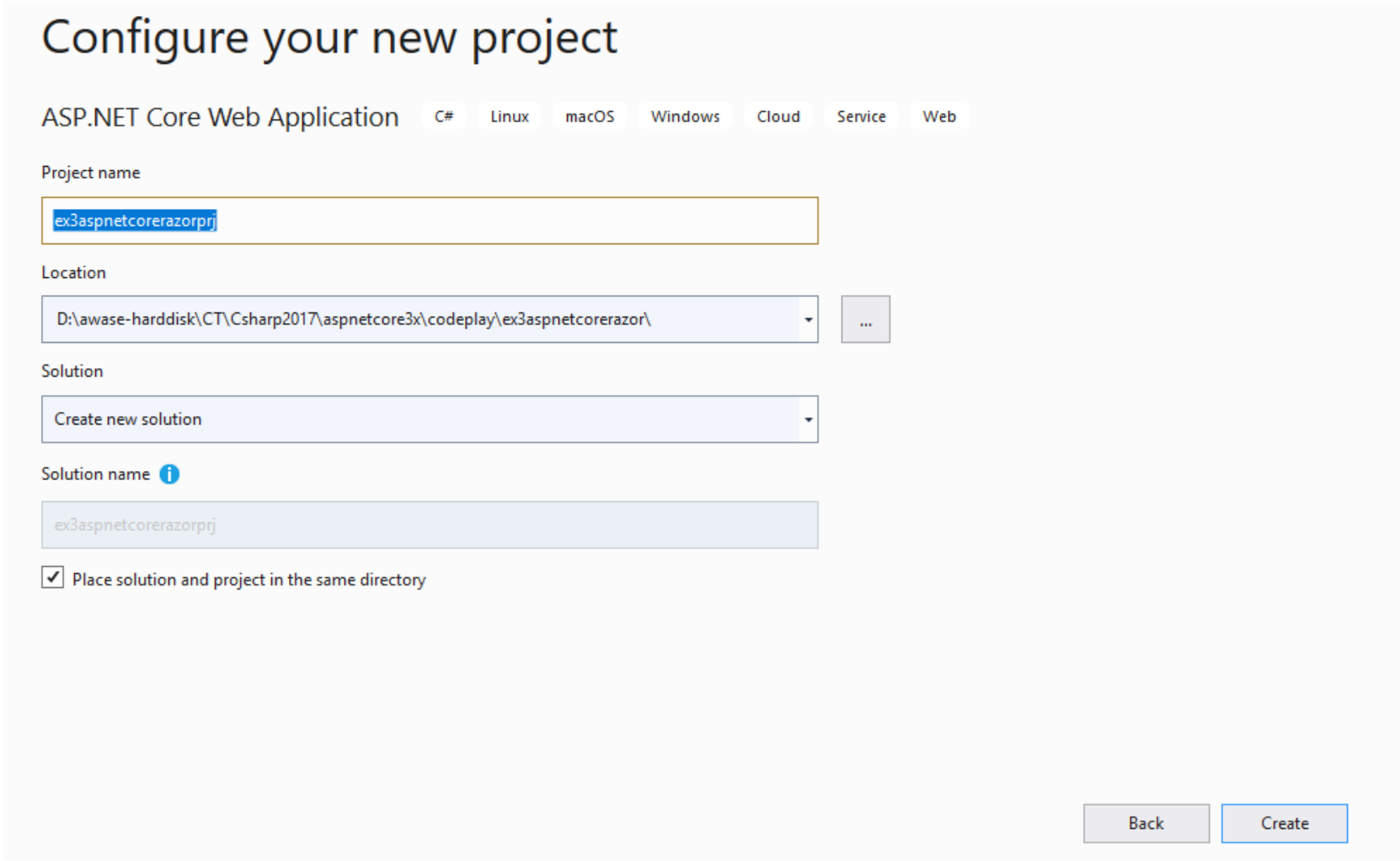
SYED AWASE KHIRNI

Example 3: ASP.NET Core Razor Web Application

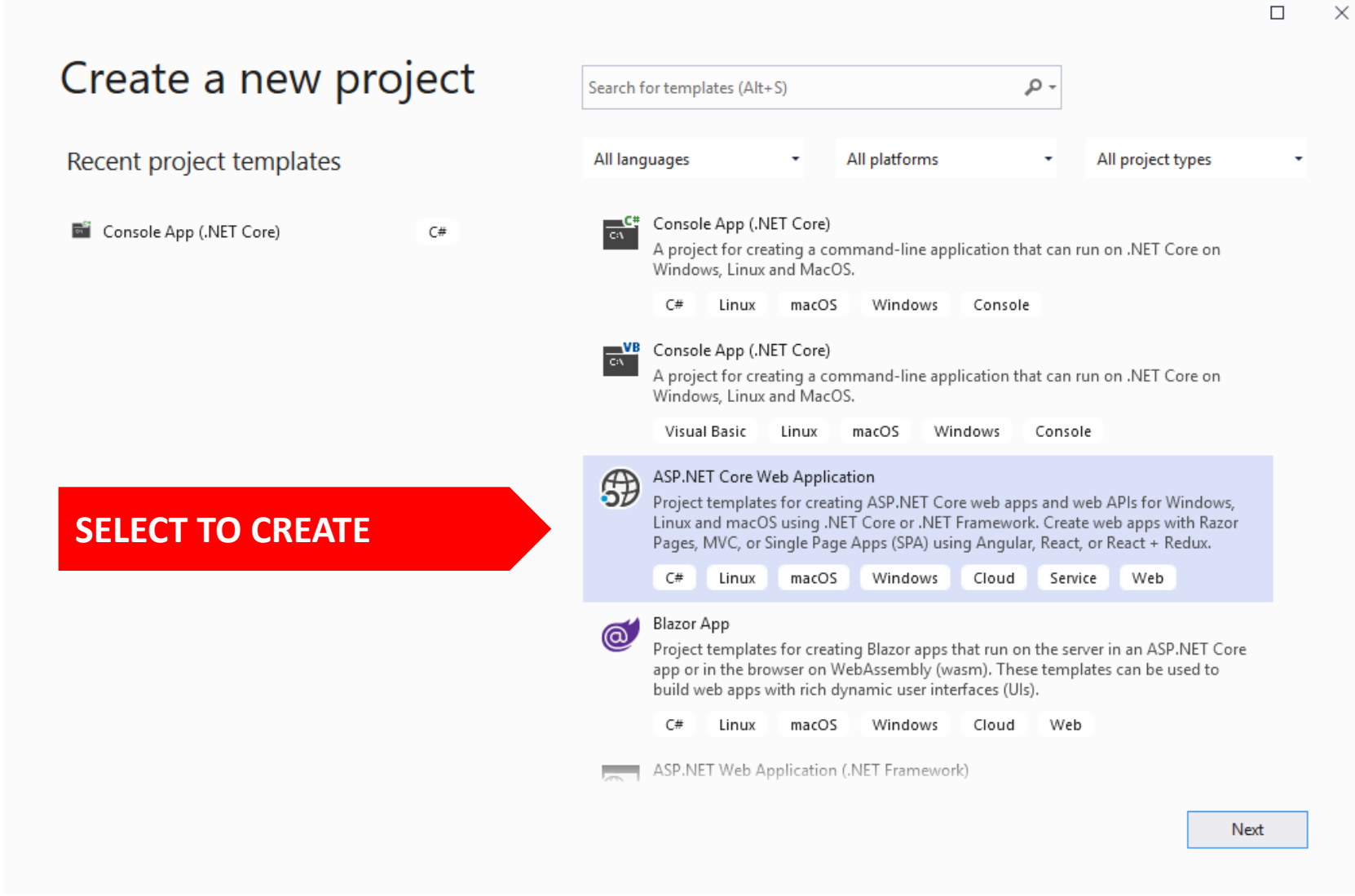
Learning Outcomes:

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**

Example: 3



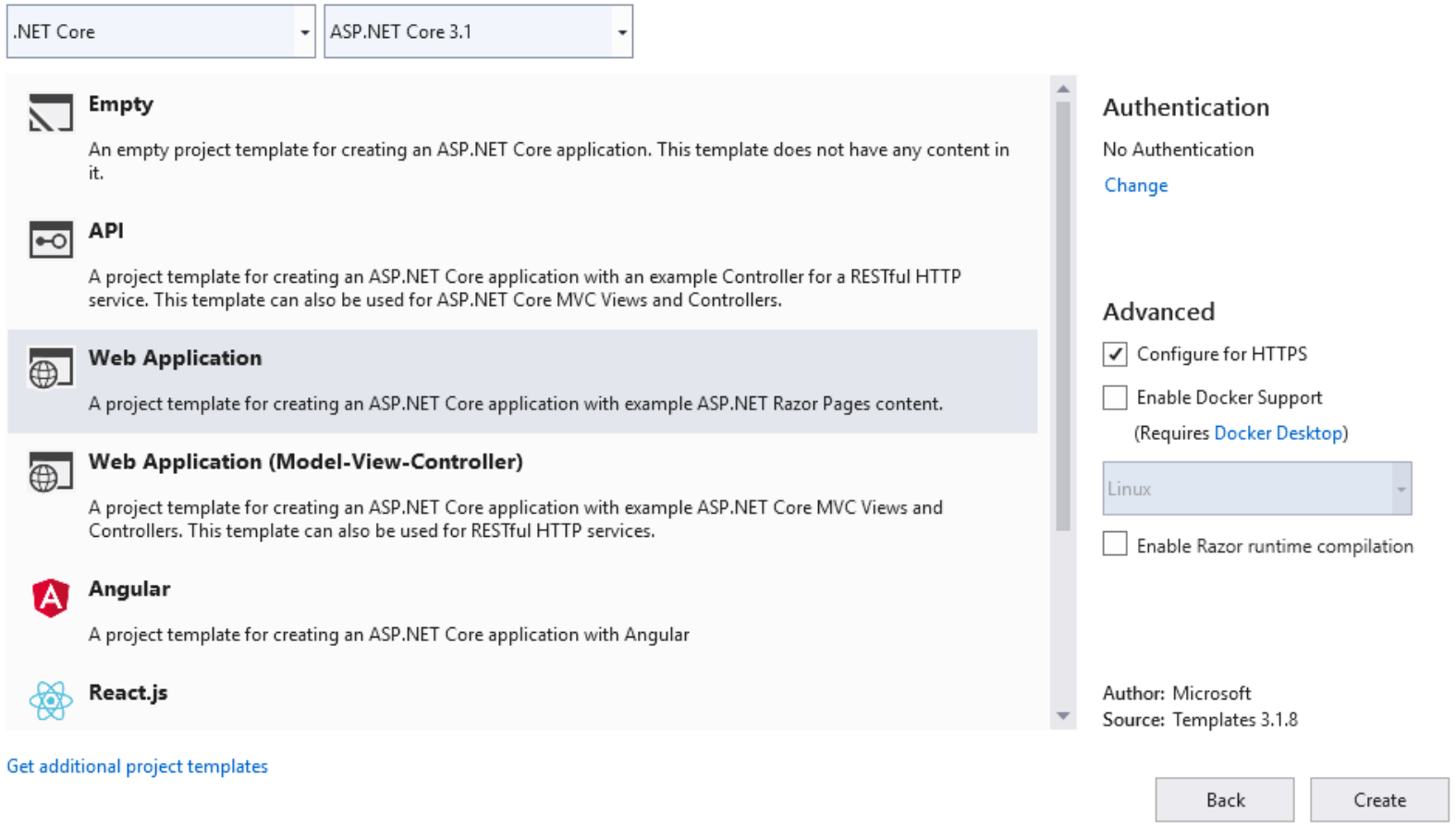
Example:



SELECT TO CREATE

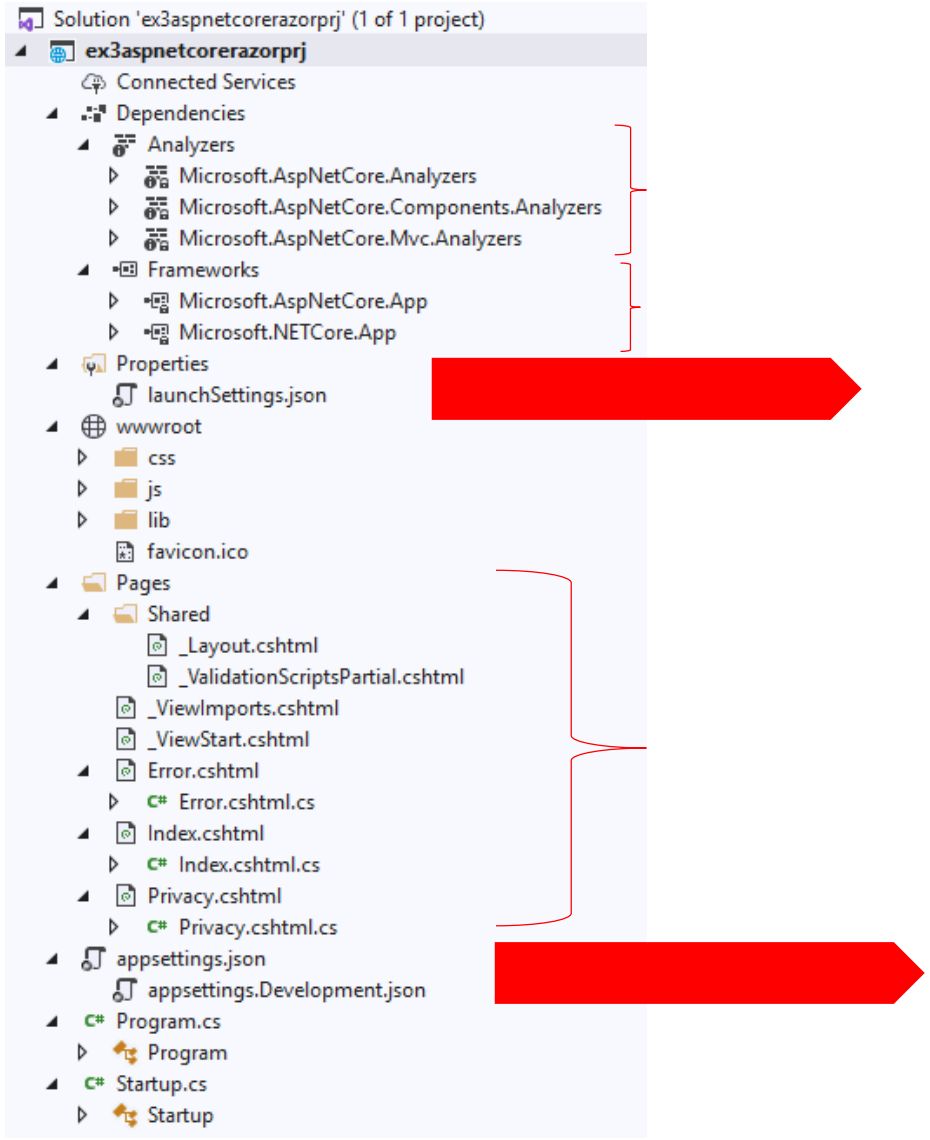
Original Series

Create a new ASP.NET Core web application



Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

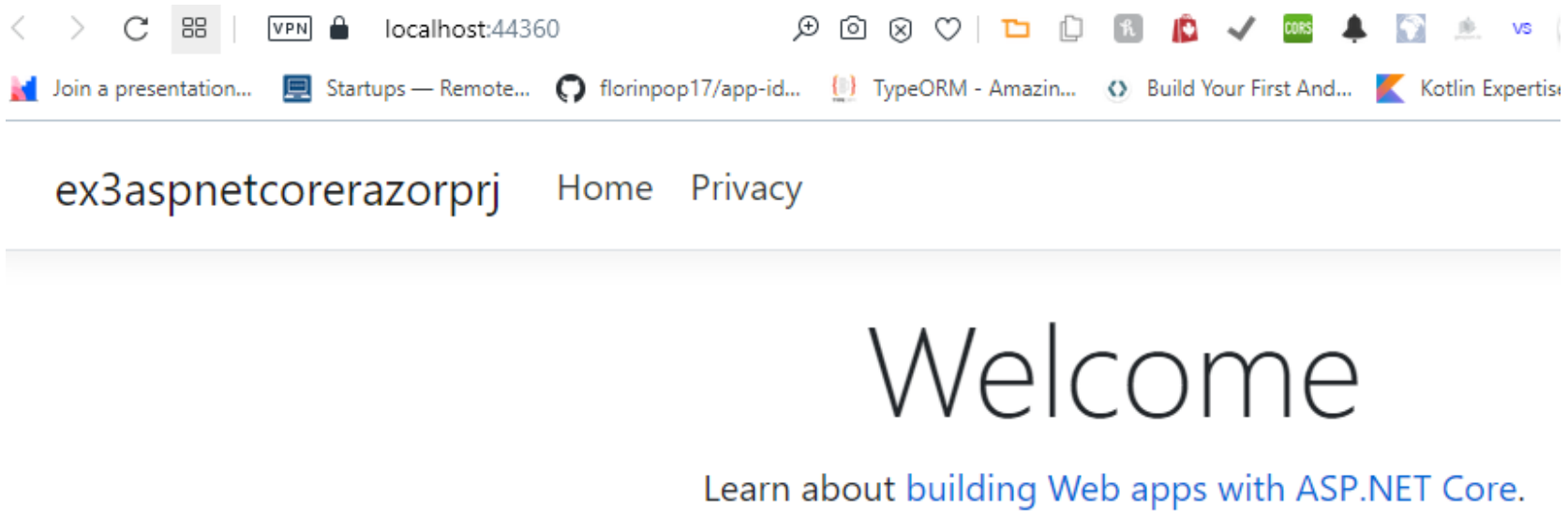
Example: 3



Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

http://localhost:44360



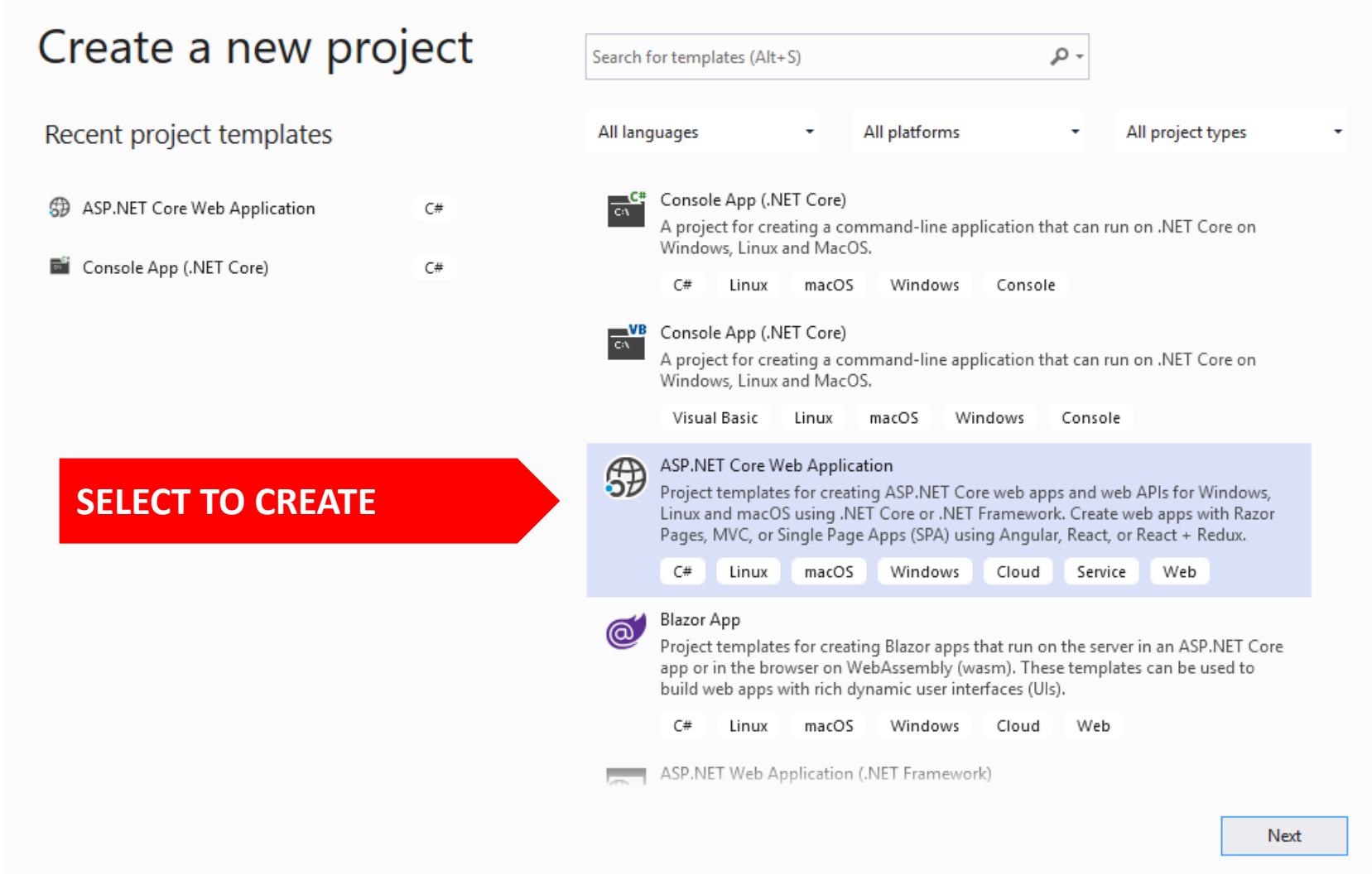
SYED AWASE KHIRNI

Example 4:ASP.NET Core Web Api Project

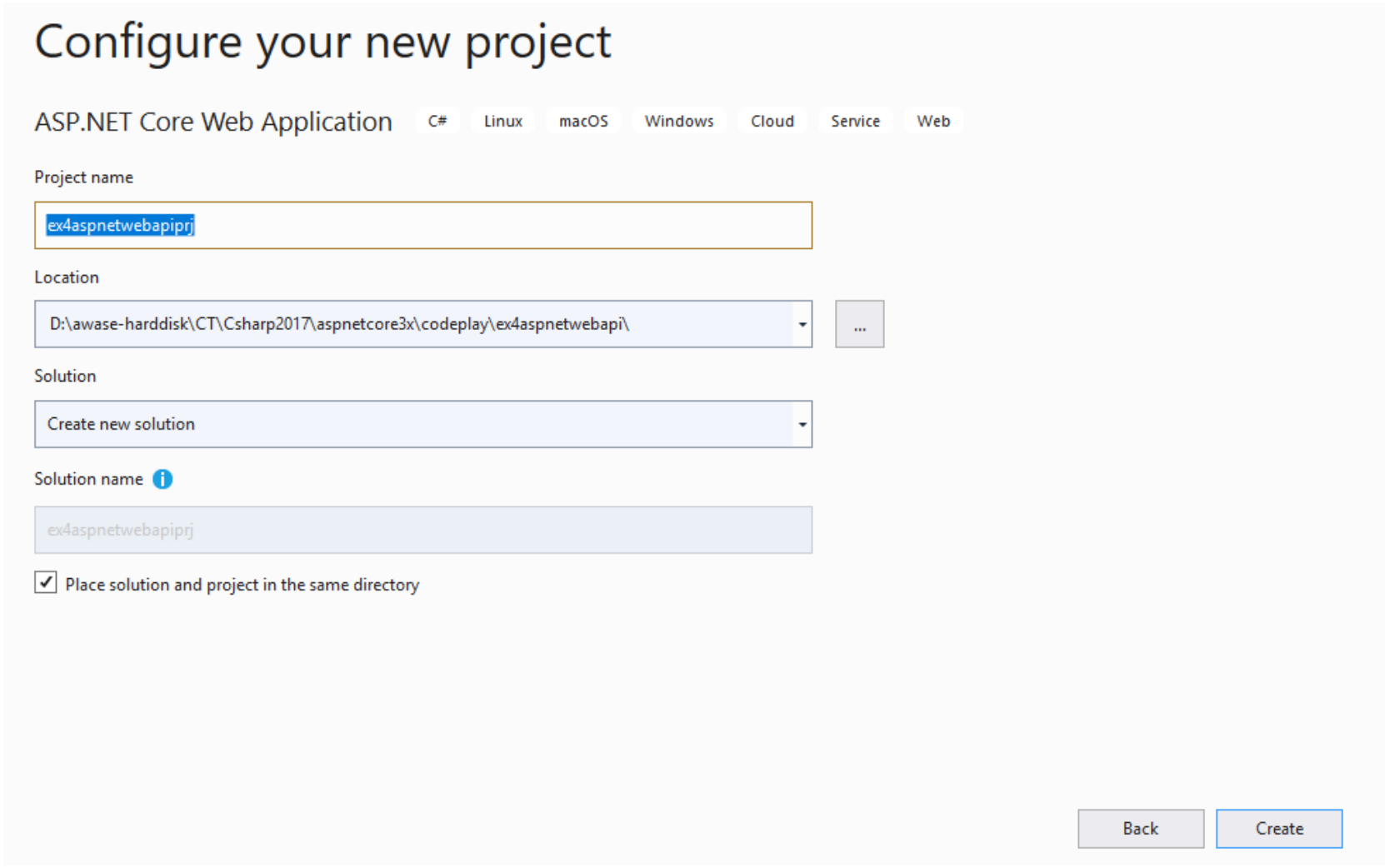
Learning Outcomes:

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**

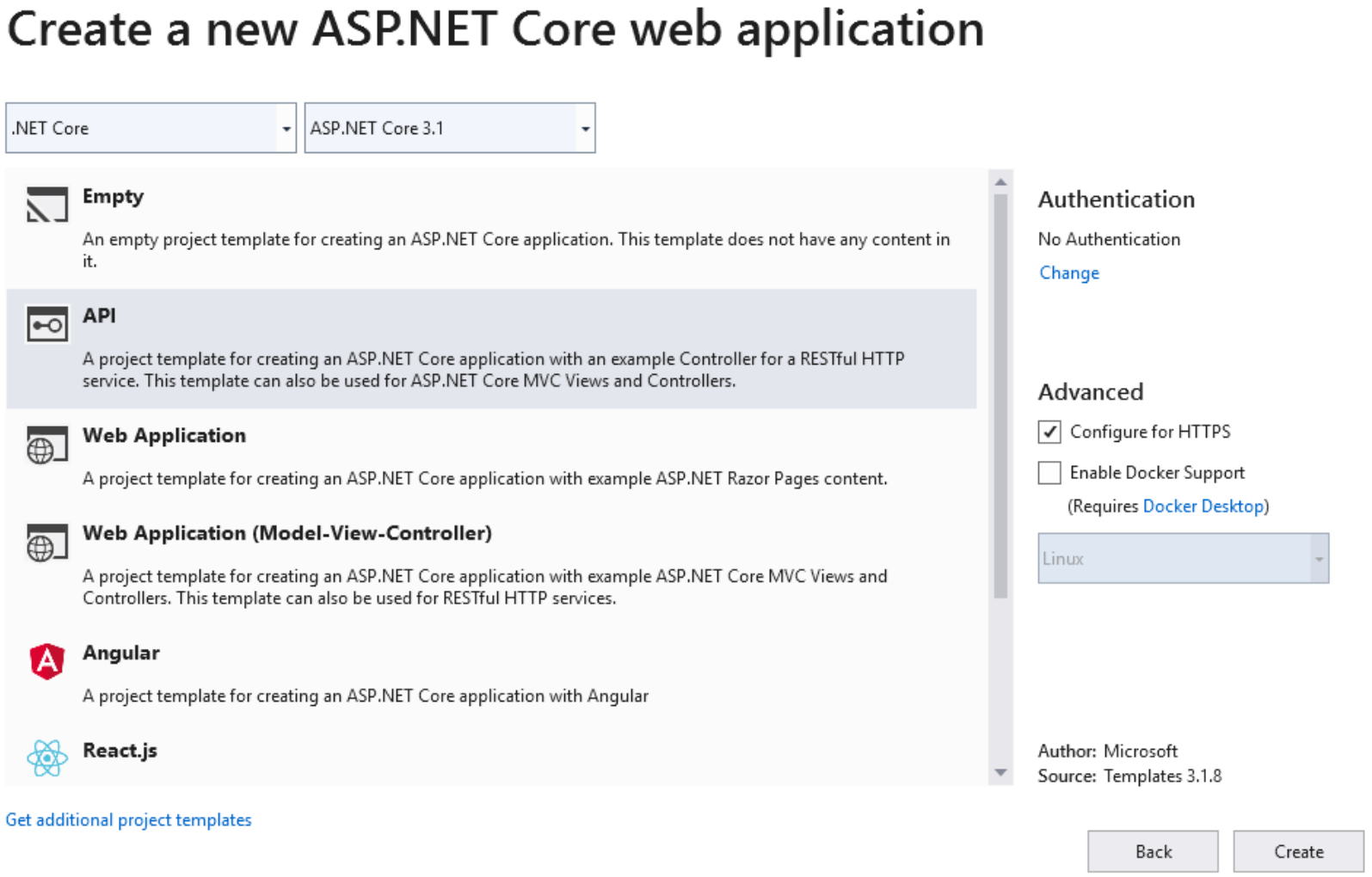
Example: 4



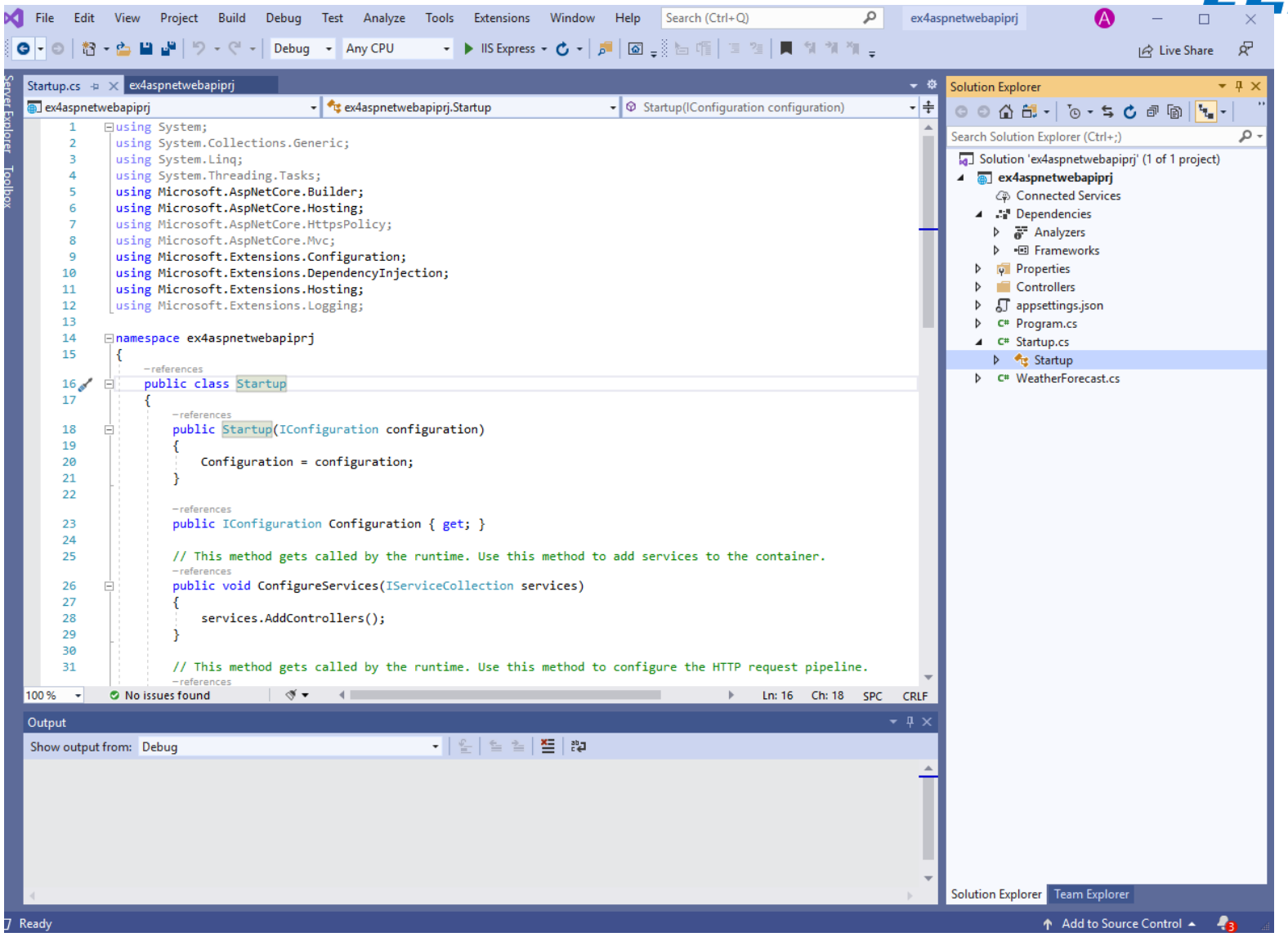
Example: 4



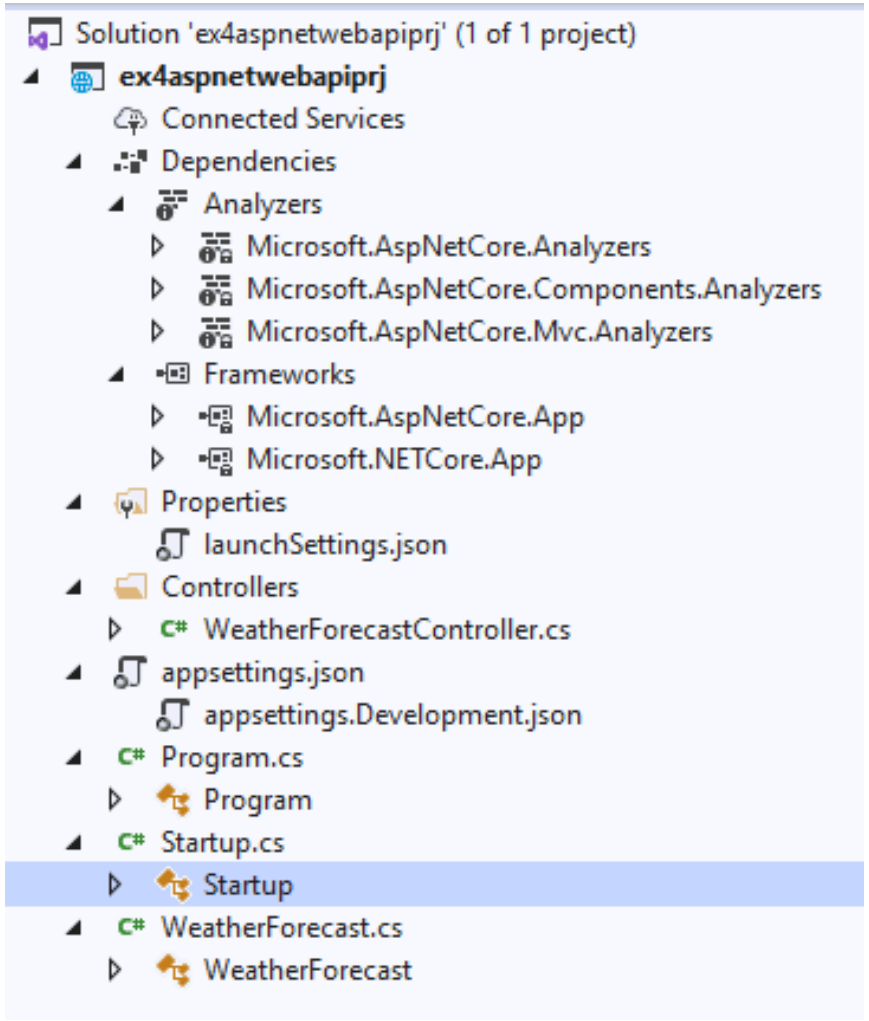
Example:



Example:

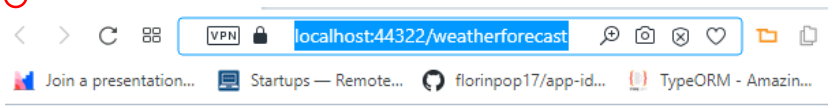


Example: 4



Example: 4

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"
  }
]
```

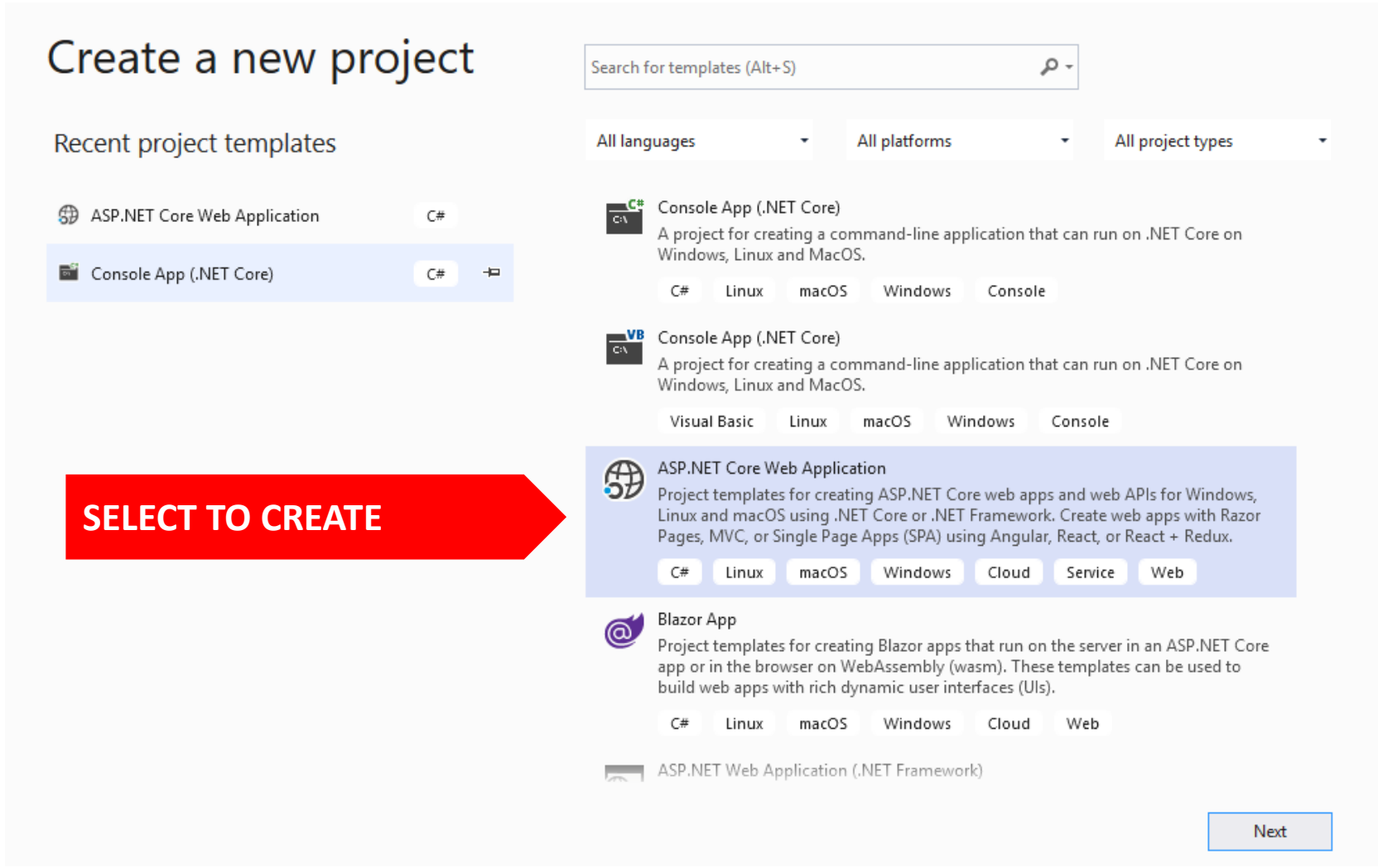
SYED AWASE KHIRNI

Example 5:ASP.NET Core MVC Web Application

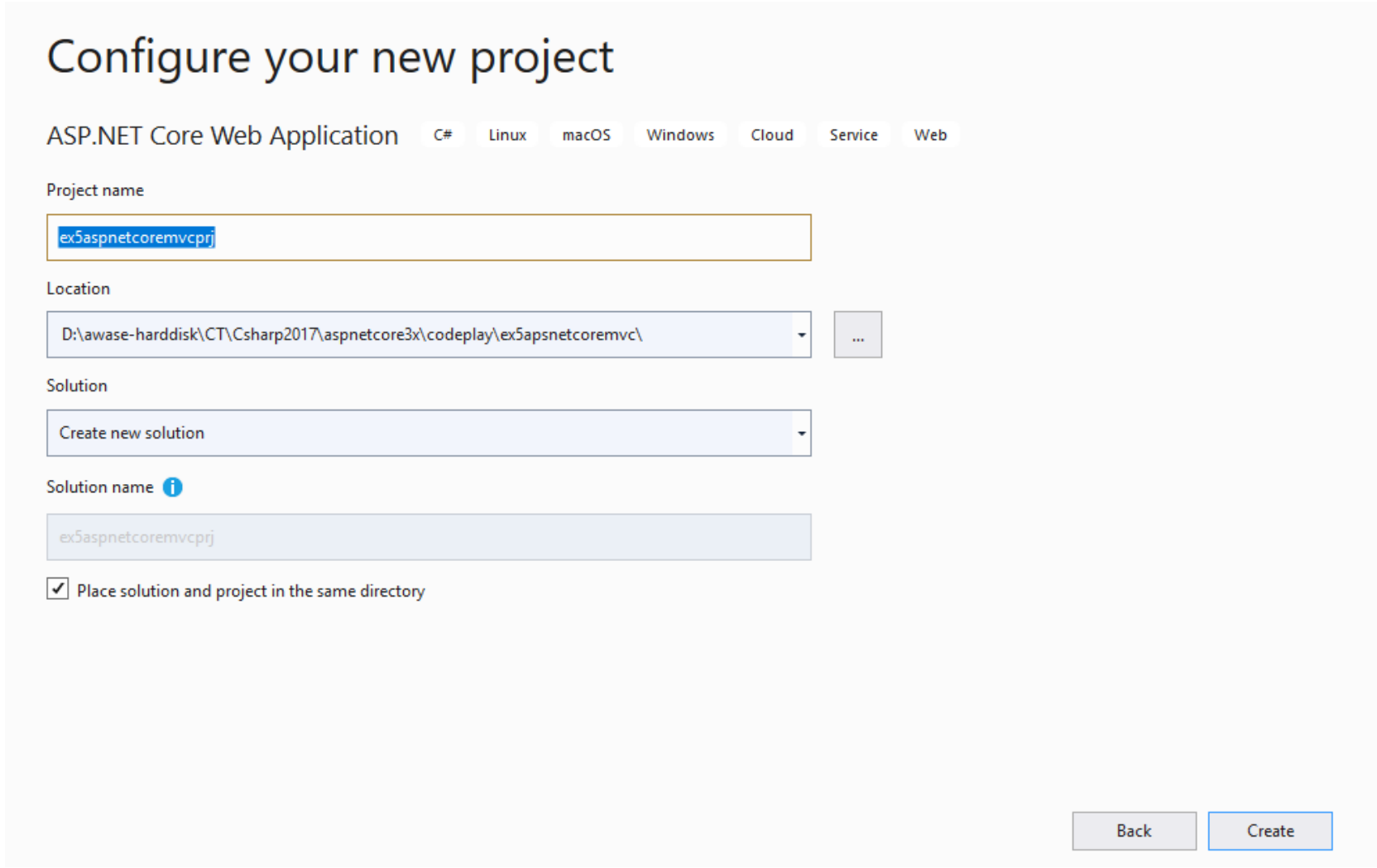
Learning Outcomes:

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**

Original Series

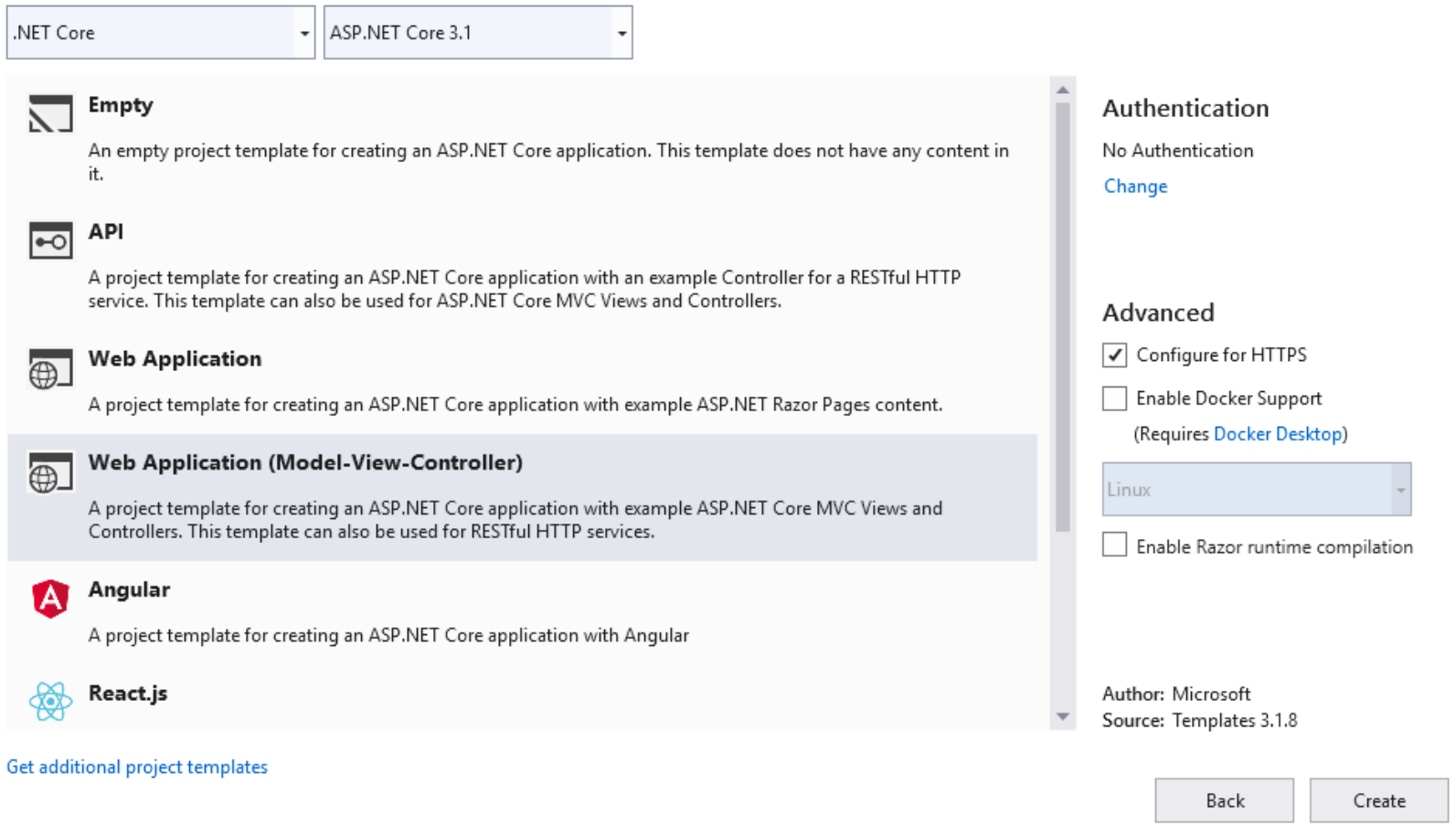


SELECT TO CREATE

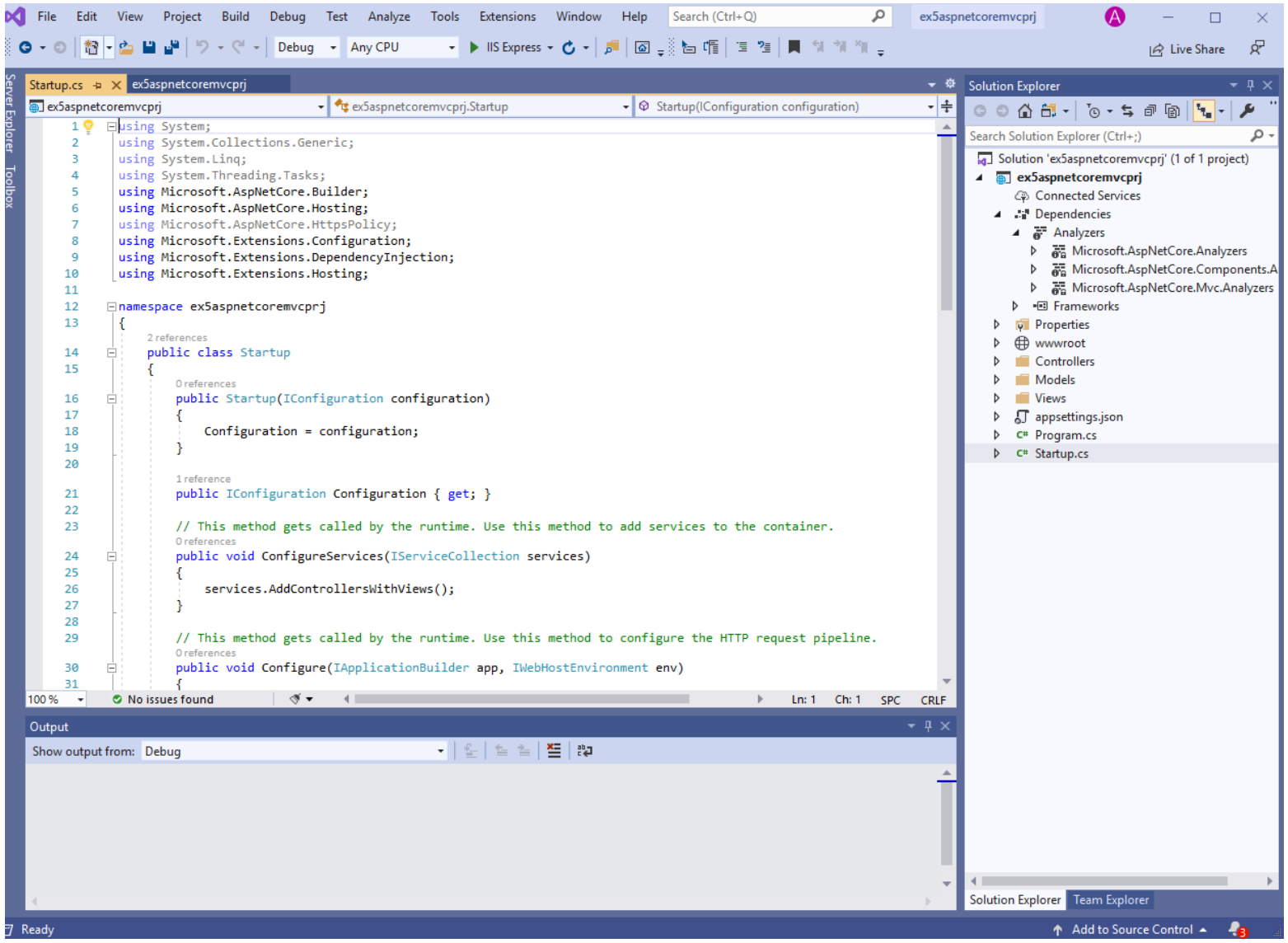


Original Series

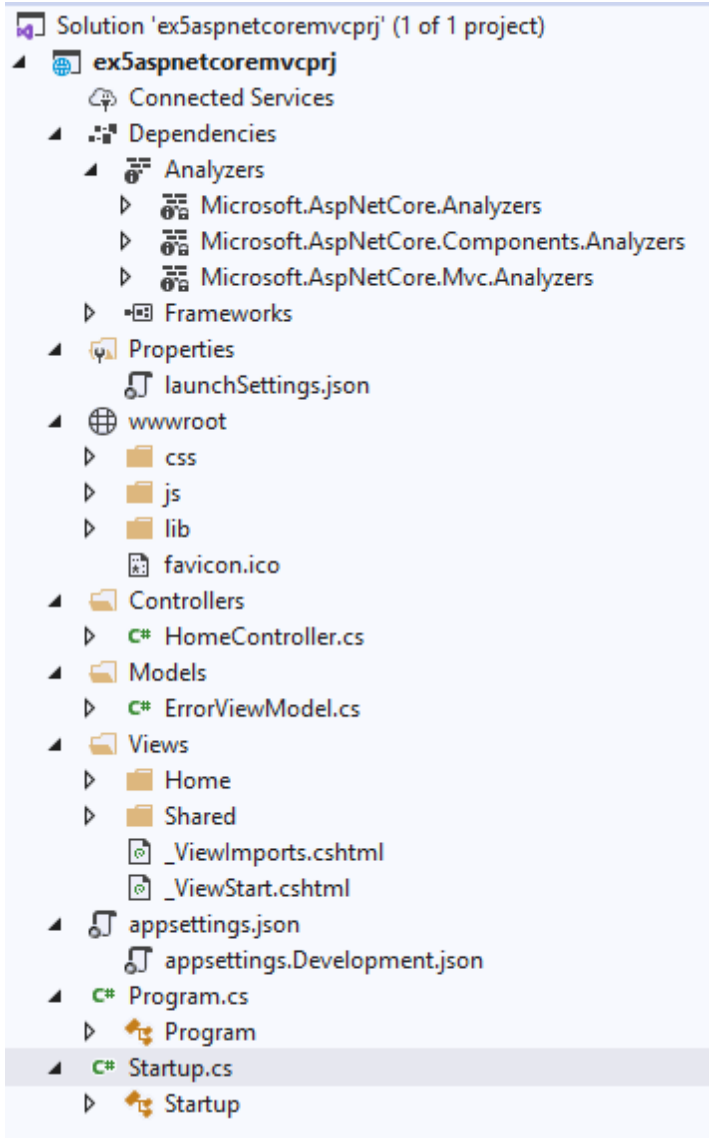
Create a new ASP.NET Core web application



Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

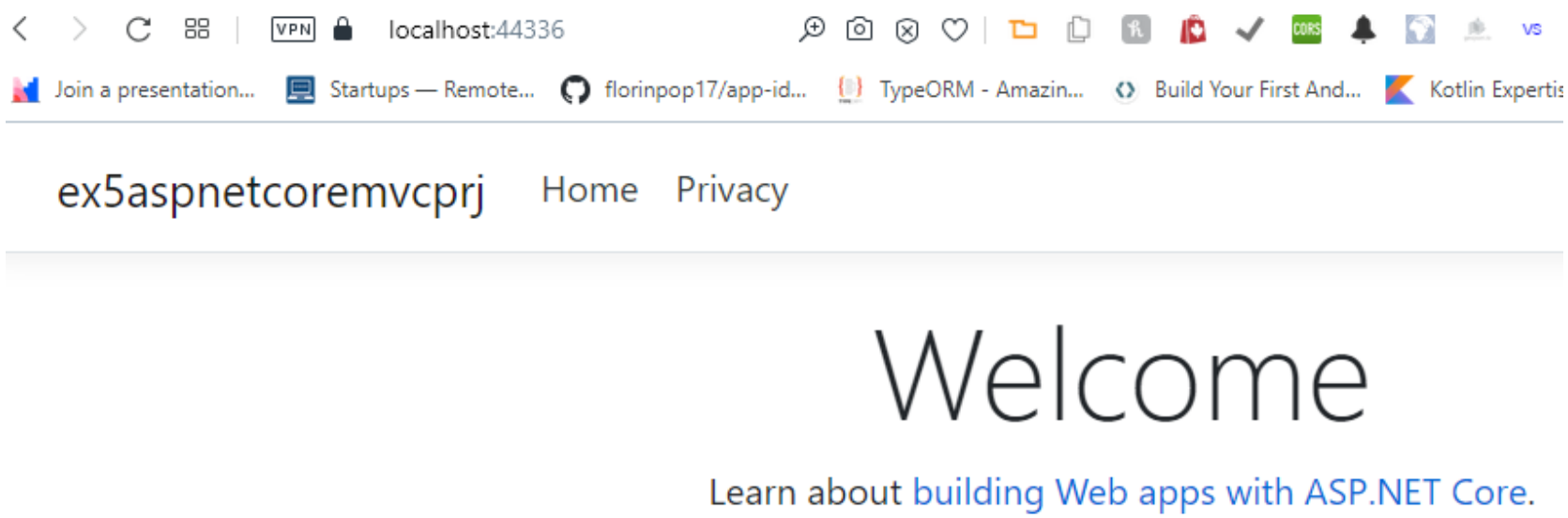


Example: 5



Example:

http://localhost:44336



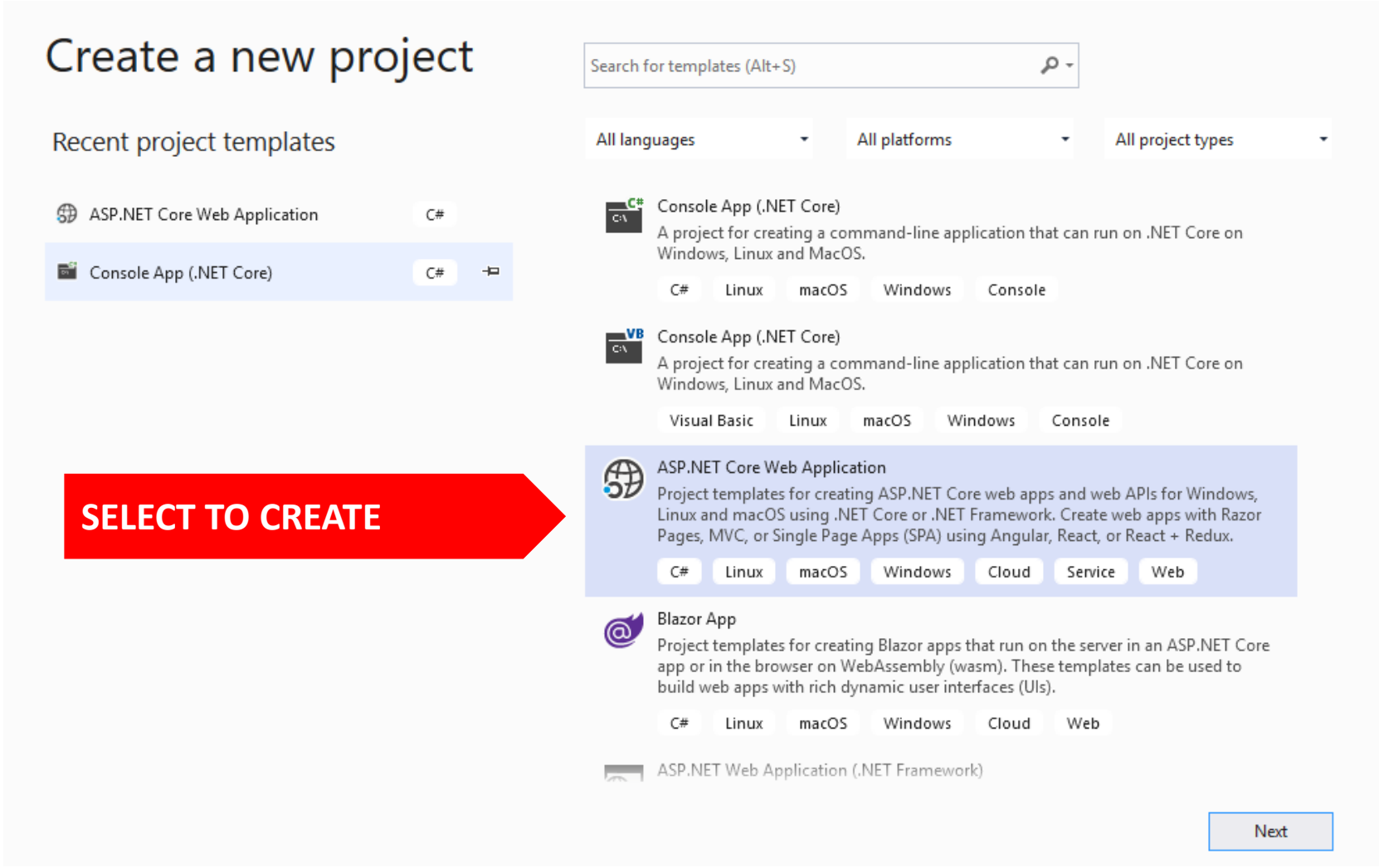
SYED AWASE KHIRNI

Example 6: ASP.NET Core Angular Stack Application

Learning Outcomes:

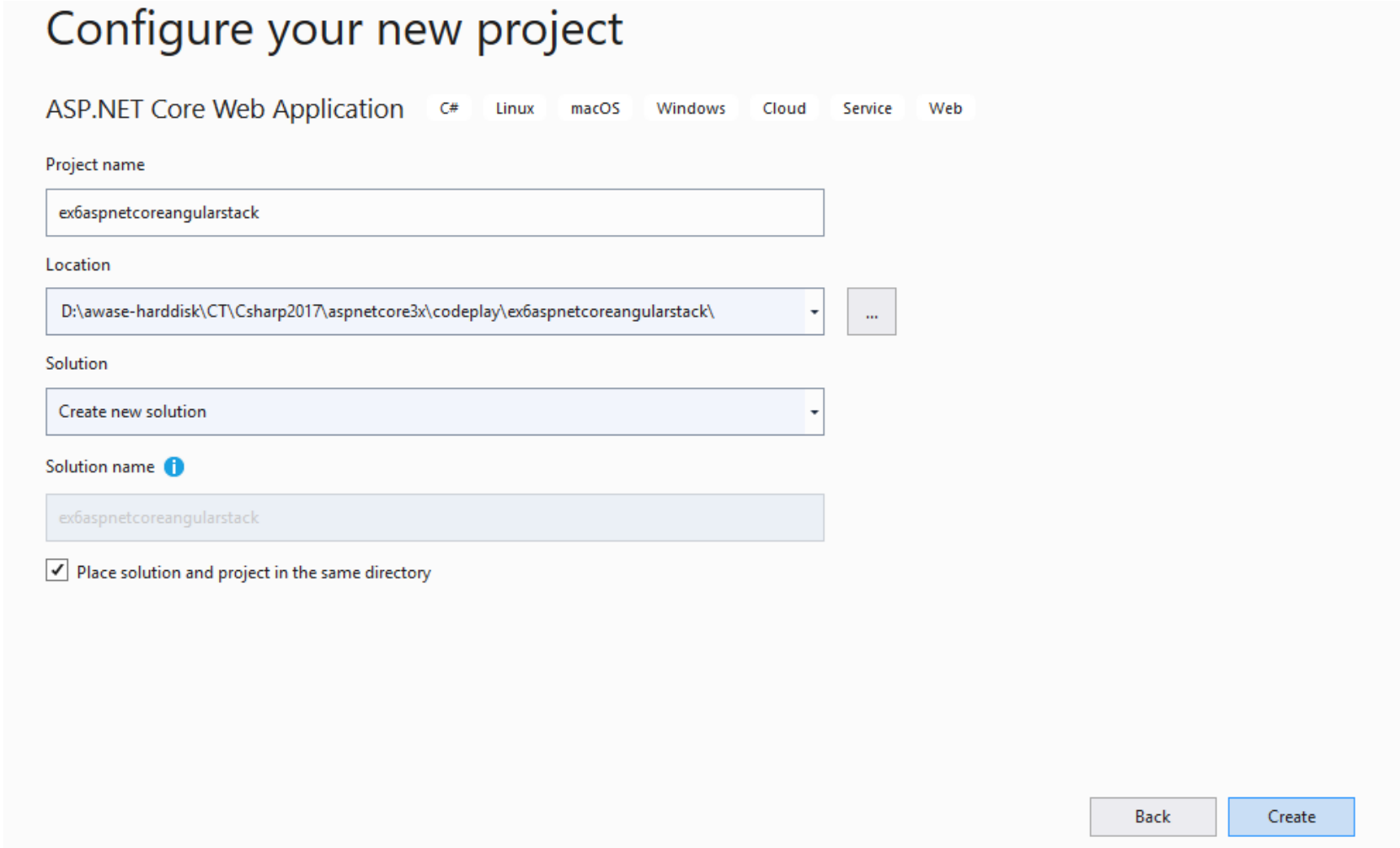
1. **Understand the steps to create ASP.NET Core 3.x angular stack application**
2. **Understand the web application structure**
3. **Understand the key dependencies for ASP.NET Core 3.x angular stack application**
4. **Executing ASP.NET Core 3.x angular stack application and rendering it on http://localhost:44360**

Example: 6



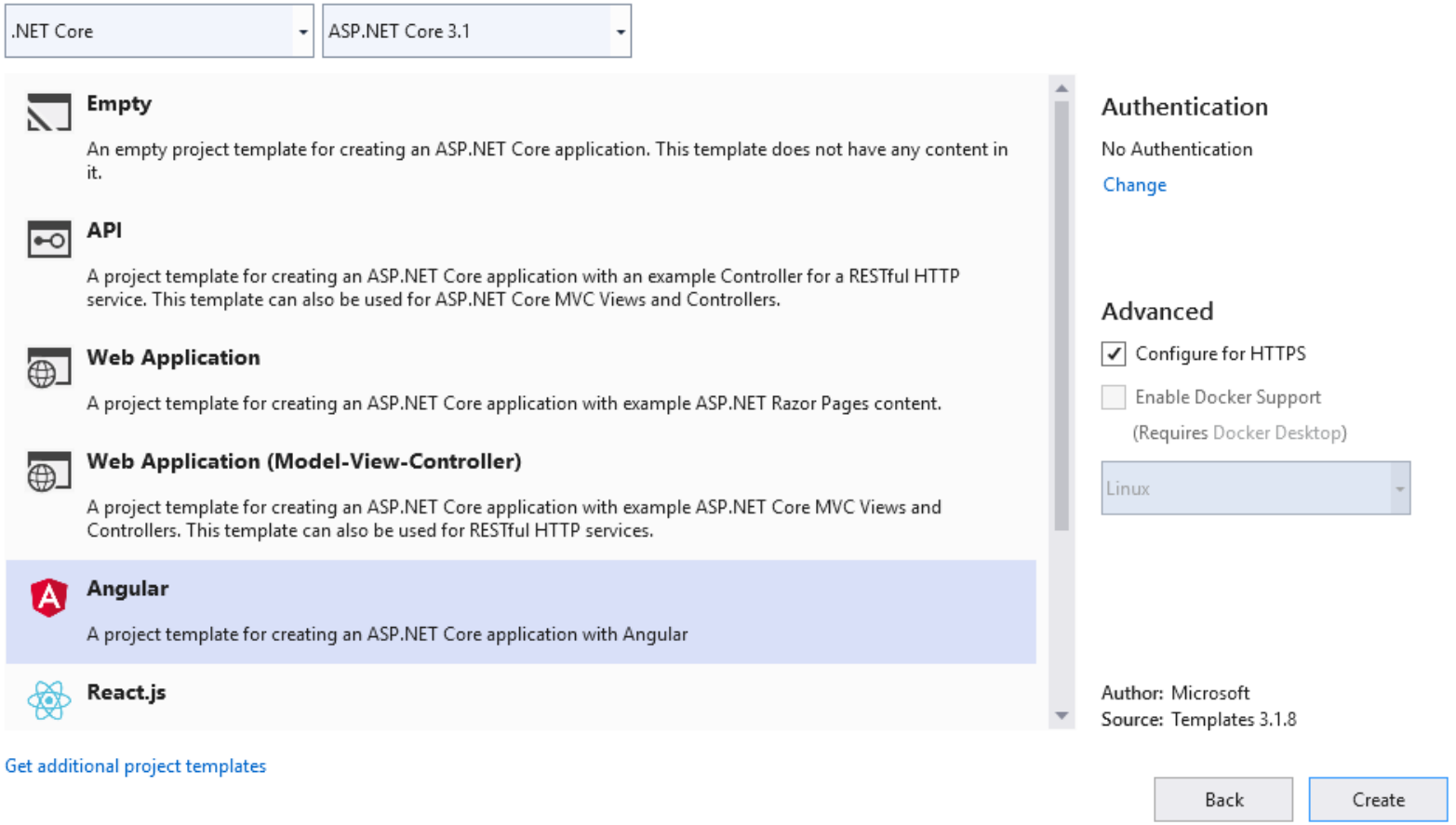
SELECT TO CREATE

Example: 6



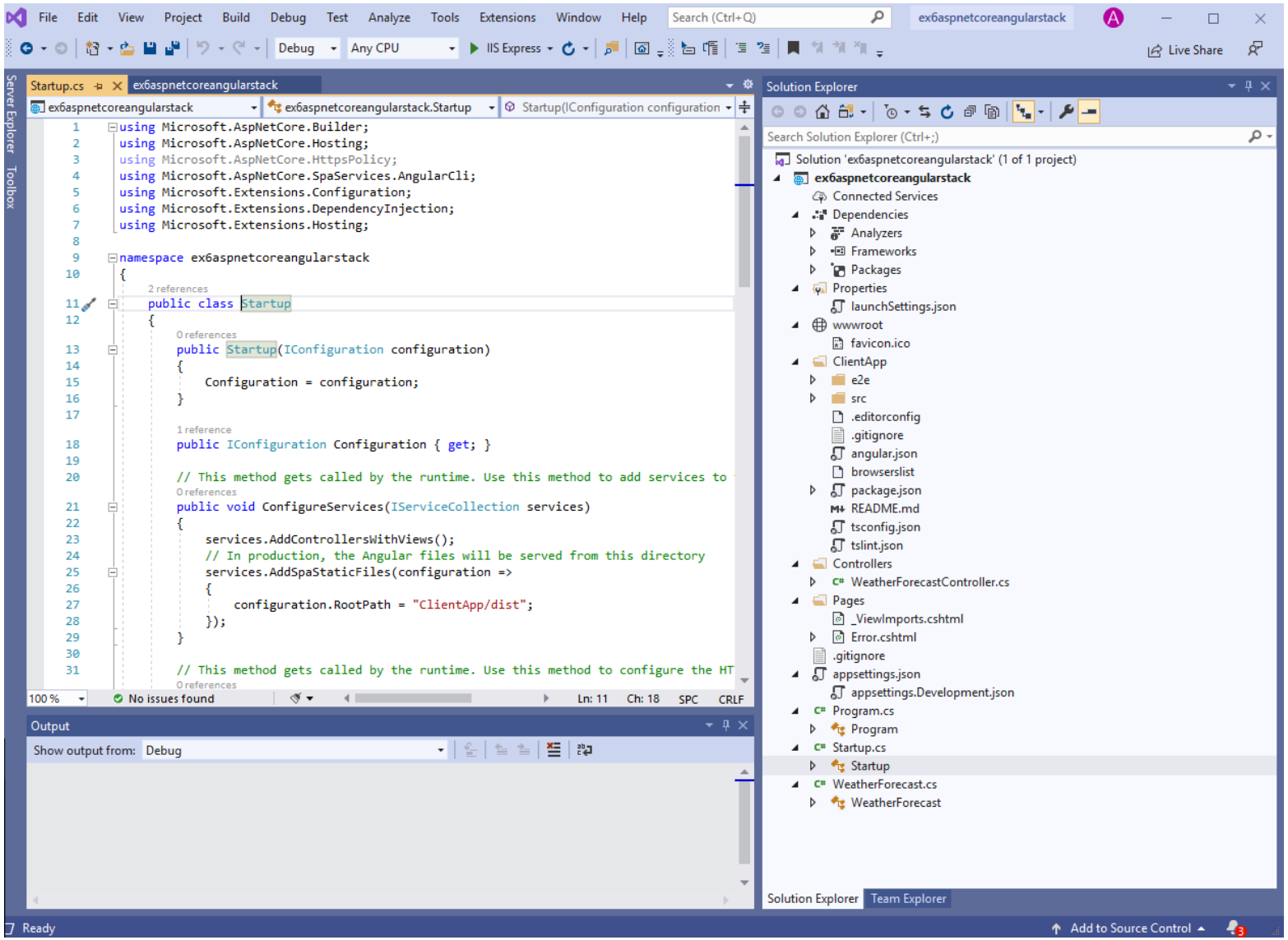
Original Series

Create a new ASP.NET Core web application



Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

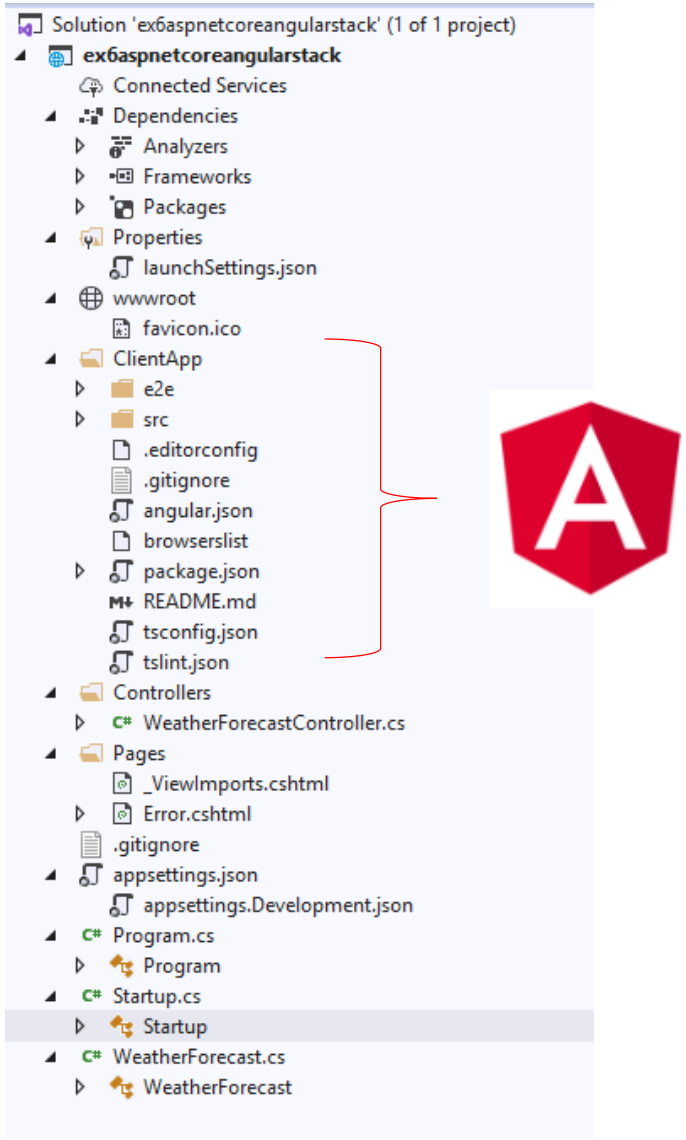
Example: 6



Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

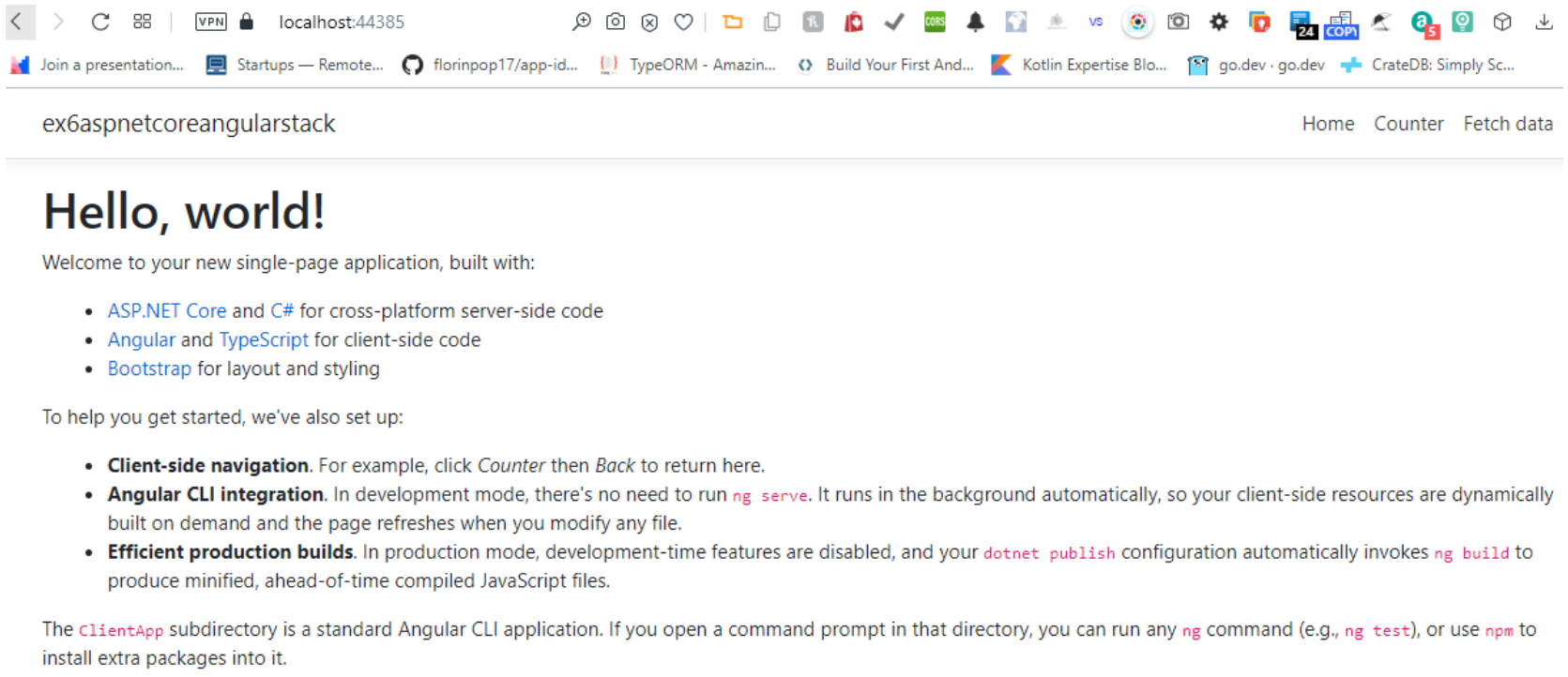
Example: 6



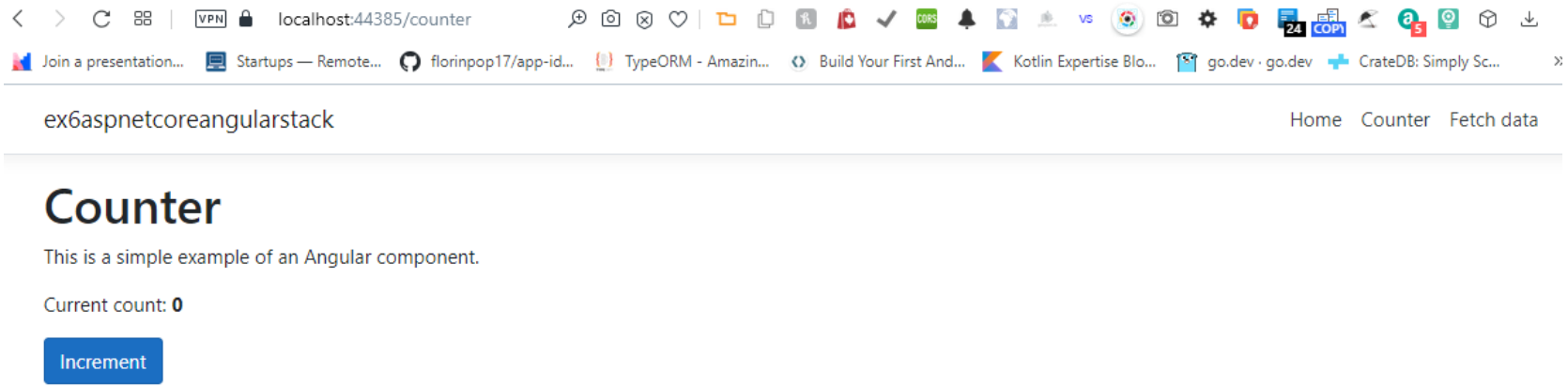
Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

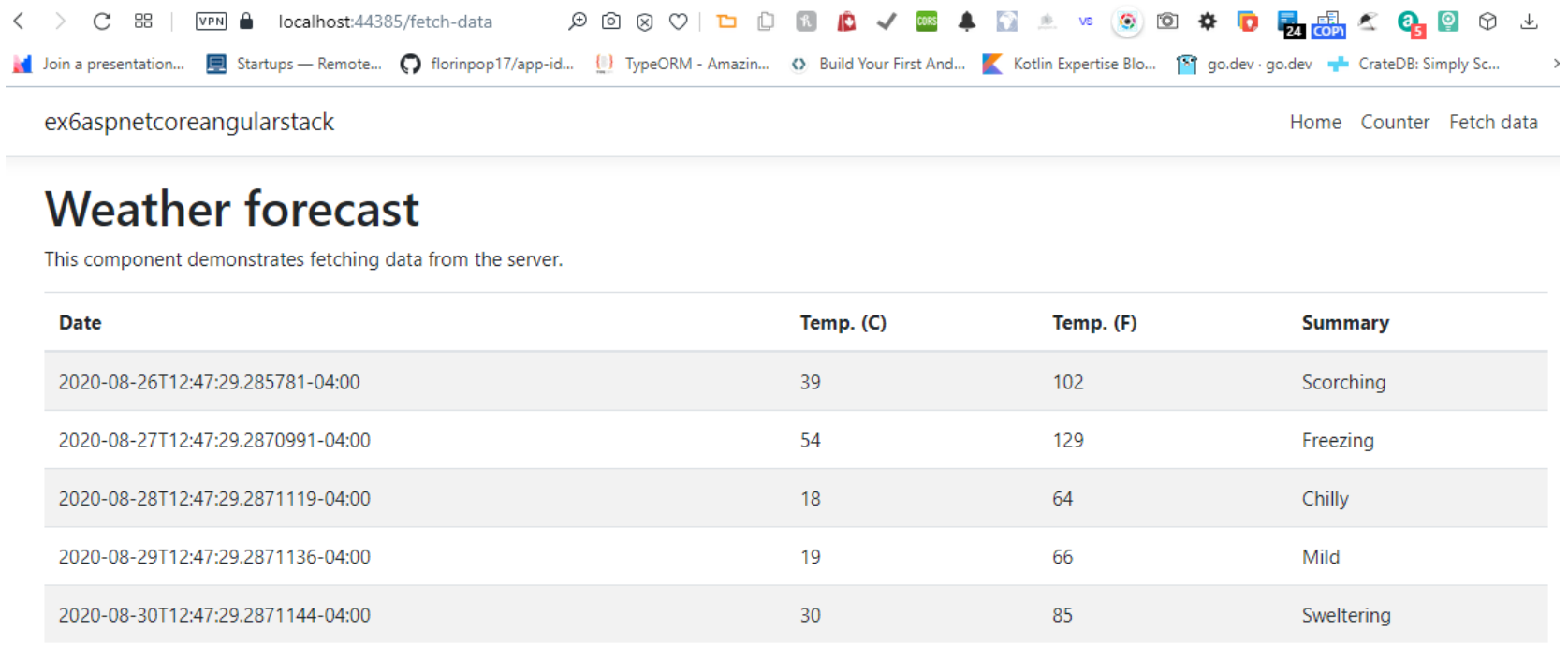
http://localhost:44385



http://localhost:44385/counter



http://localhost:44385/fetch-data



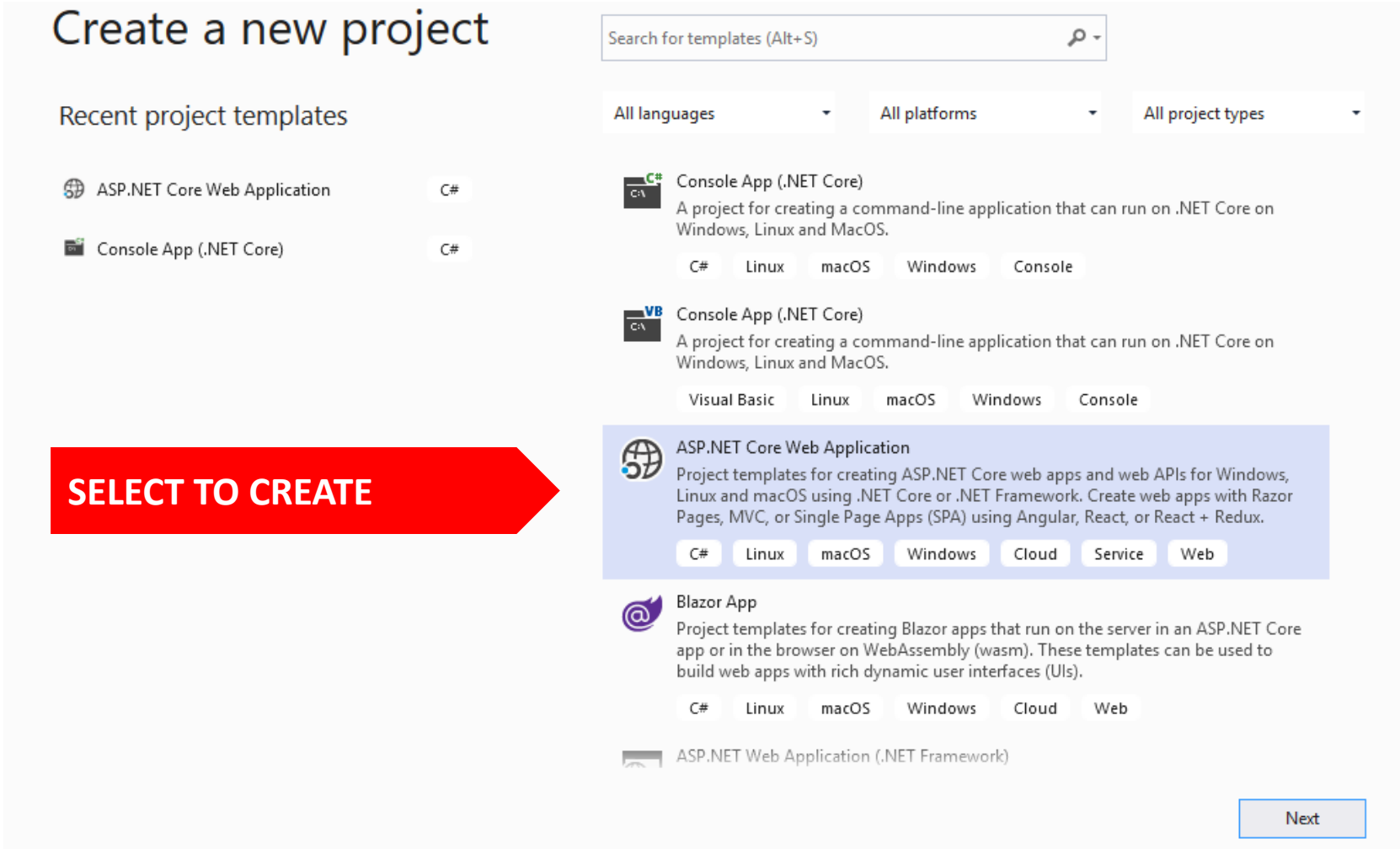
SYED AWASE KHIRNI

Example 7: ASP.NET Core React Stack Application

Learning Outcomes:

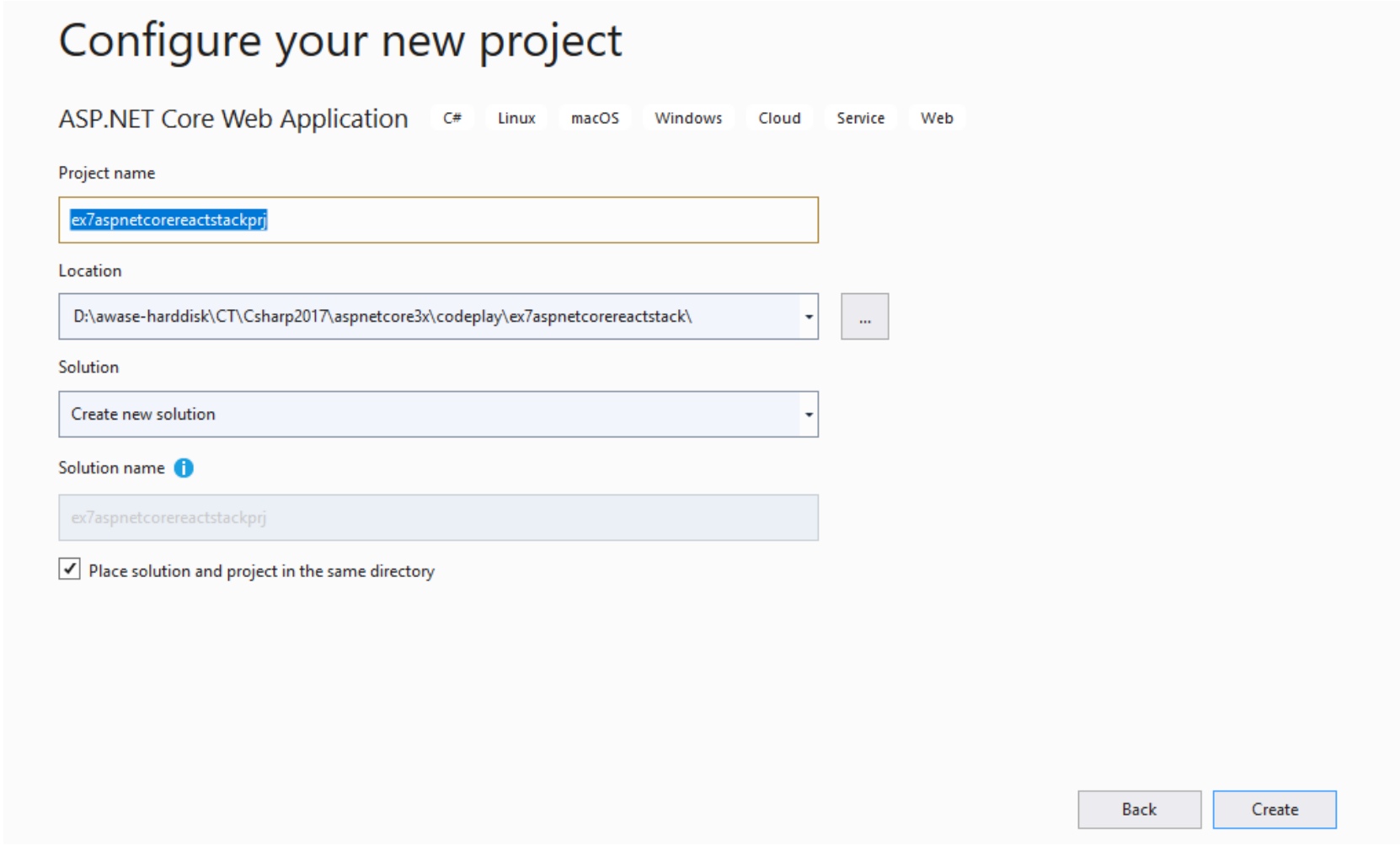
1. **Understand the steps to create ASP.NET Core 3.x react stack application**
2. **Understand the web application structure**
3. **Understand the key dependencies for ASP.NET Core 3.x react stack application**
4. **Executing ASP.NET Core 3.x react stack application and rendering it on http://localhost:44360**

Example: 7

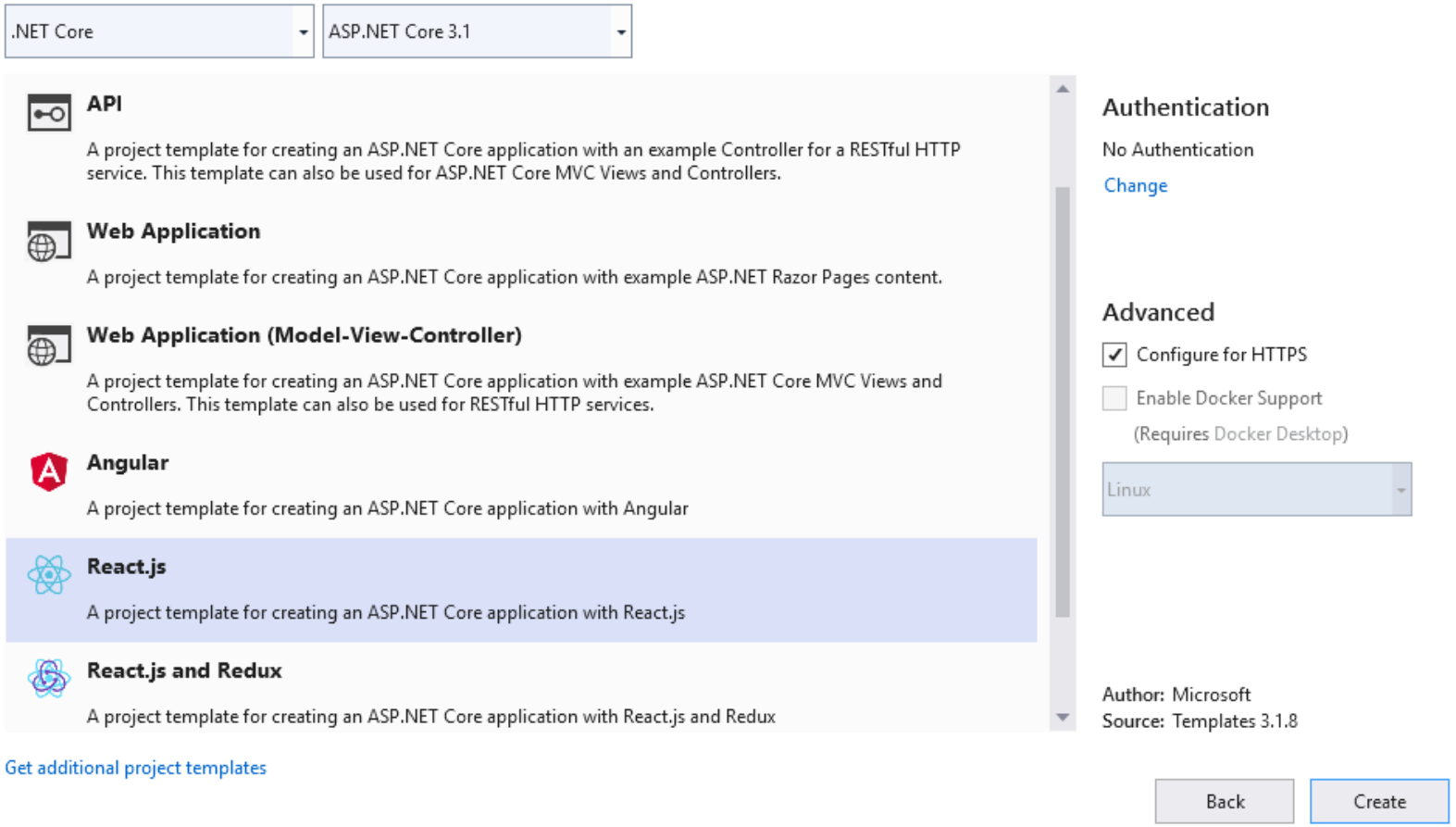


SELECT TO CREATE

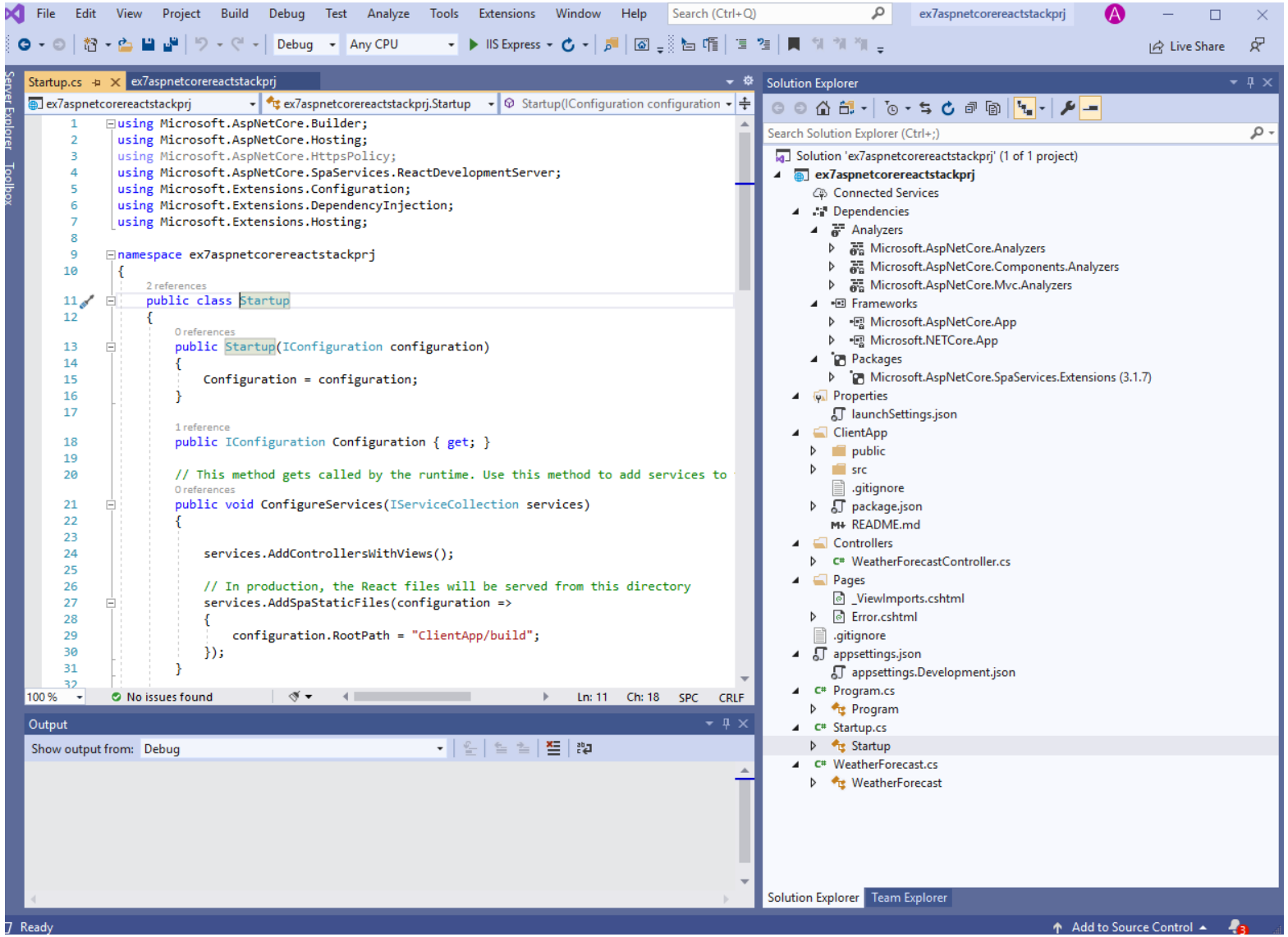
Original Series



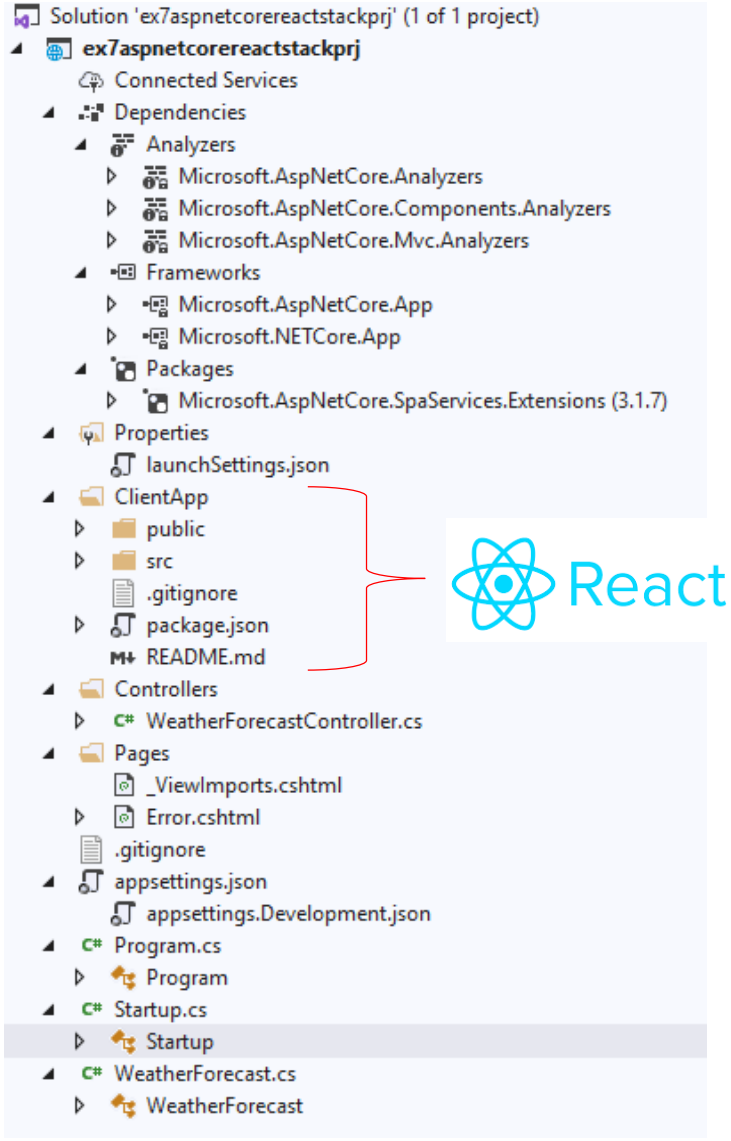
Create a new ASP.NET Core web application



Example: 7

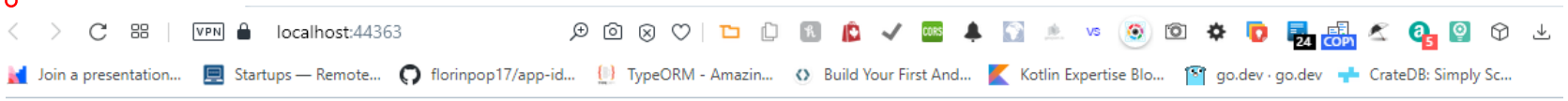


Example: 7



Example:

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

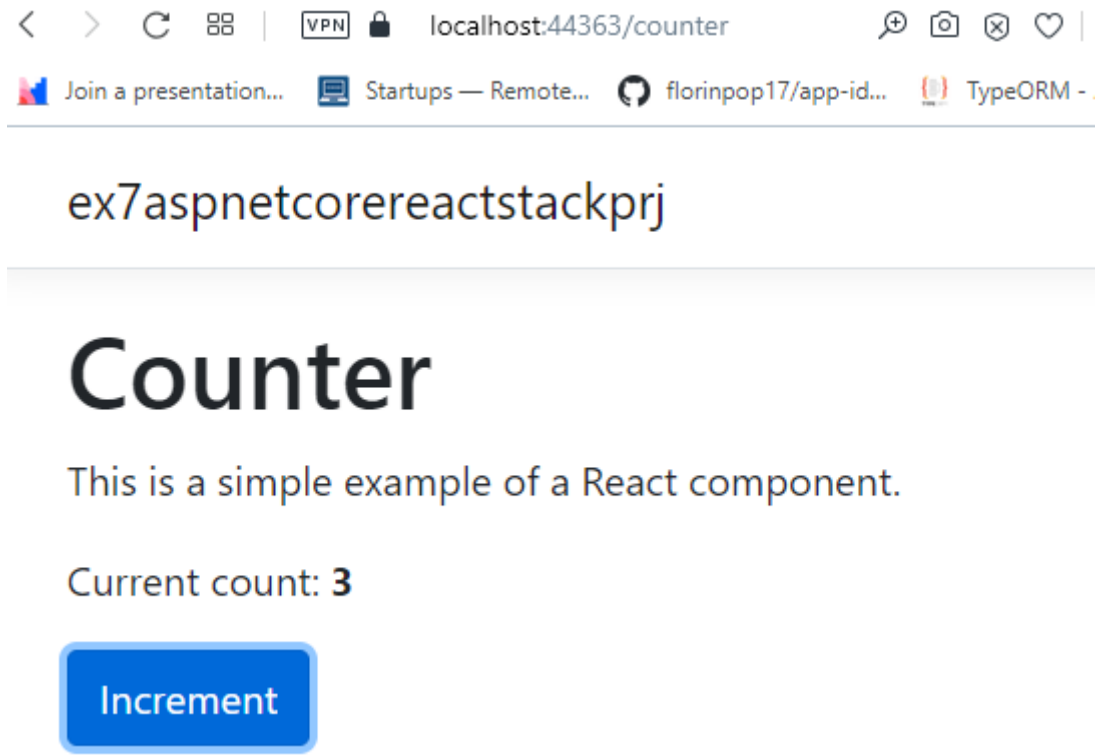
To help you get started, we have also set up:

- **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`.

Example:

http://localhost:44363/counter



Original Series

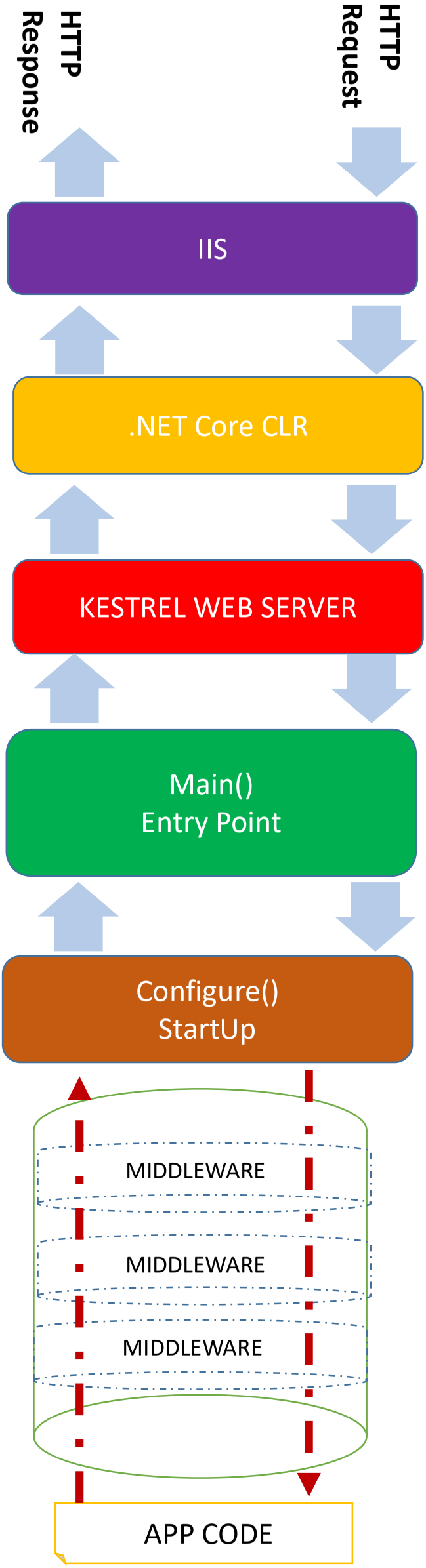
Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

SYED AWASE KHIRNI

ASP.NET Core 3.0 MVC Request Life Cycle

Learning Outcomes:

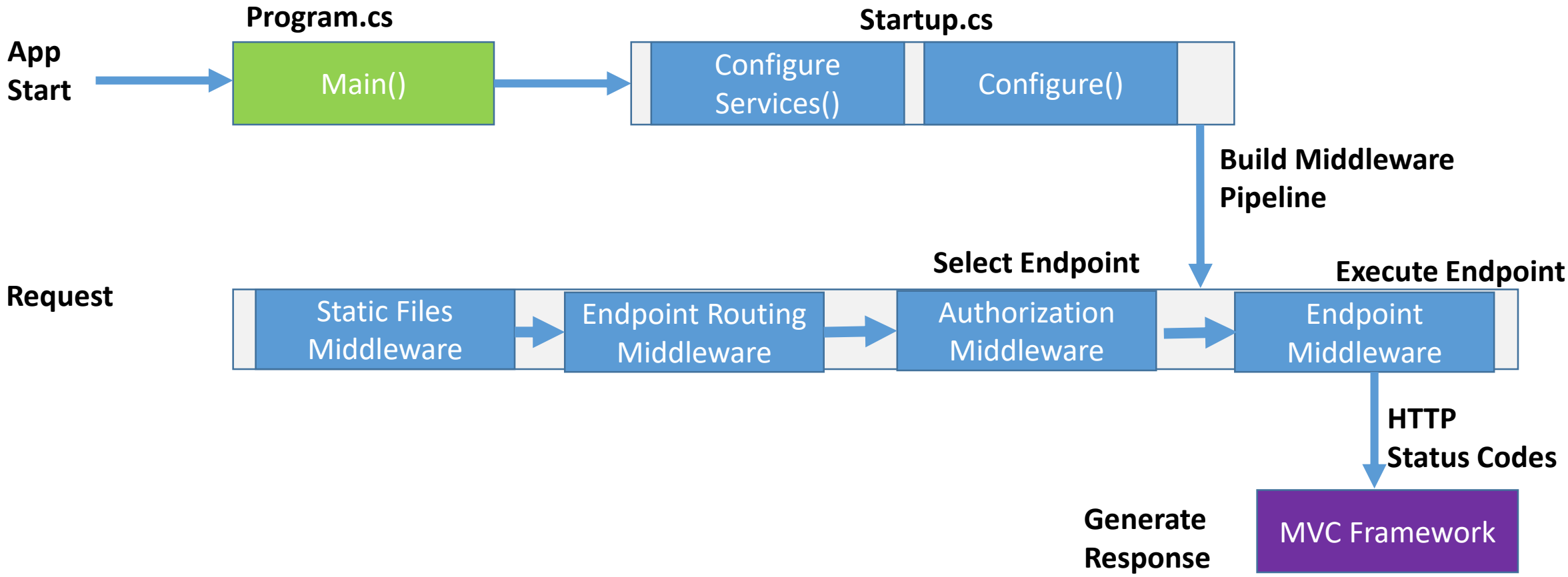
1. **Refer to ASP.NET Core 2.x Slides about the evolution of web architectures and MVC foundations https://github.com/awasekhirni/.NetCORE**
2. **Learning Outcome 2**



Updated on May 2020

https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

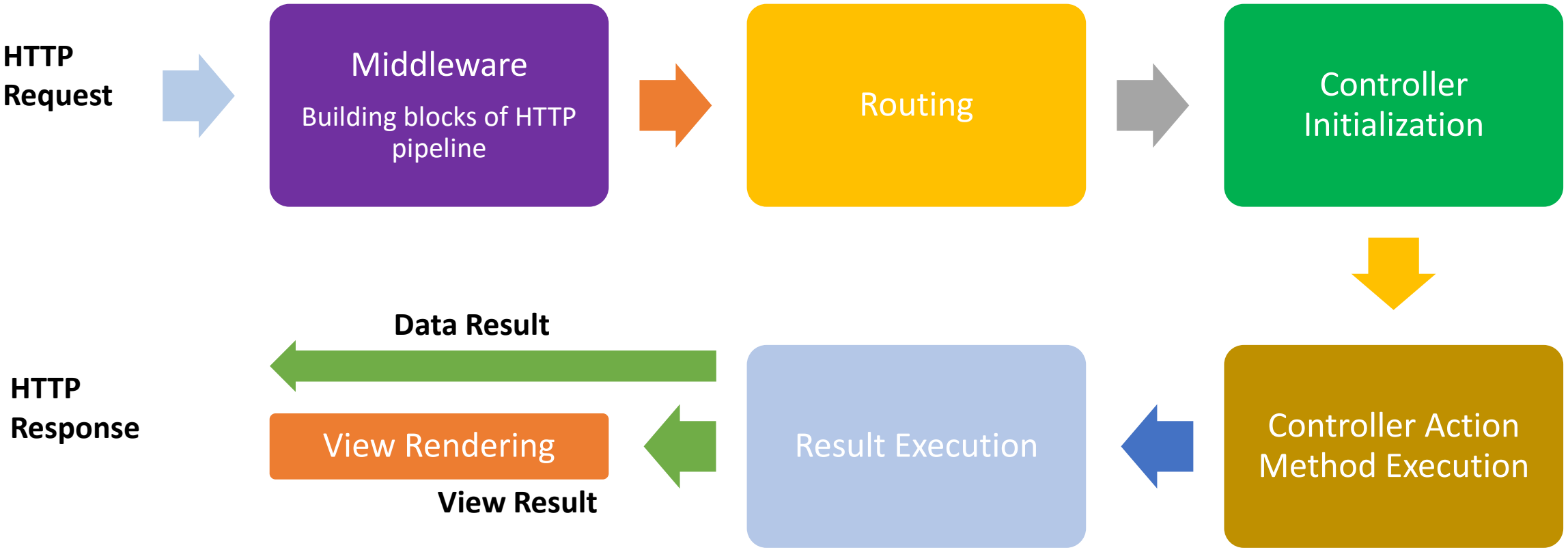
MVC Middleware Pipeline



Original Series

Updated on May 2020 https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

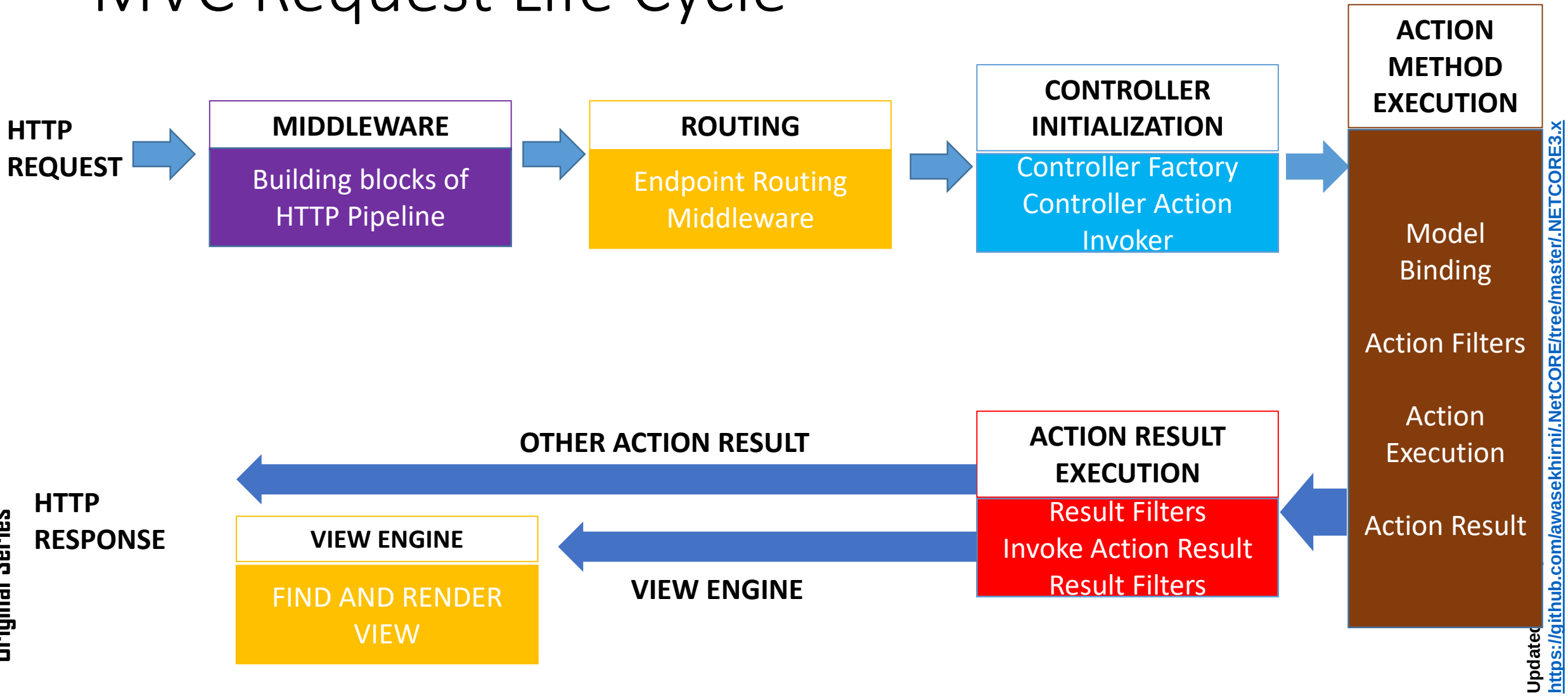
MVC Request Life Cycle



Original Series

Updated on May 2020 https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

MVC Request Life Cycle



Original Series

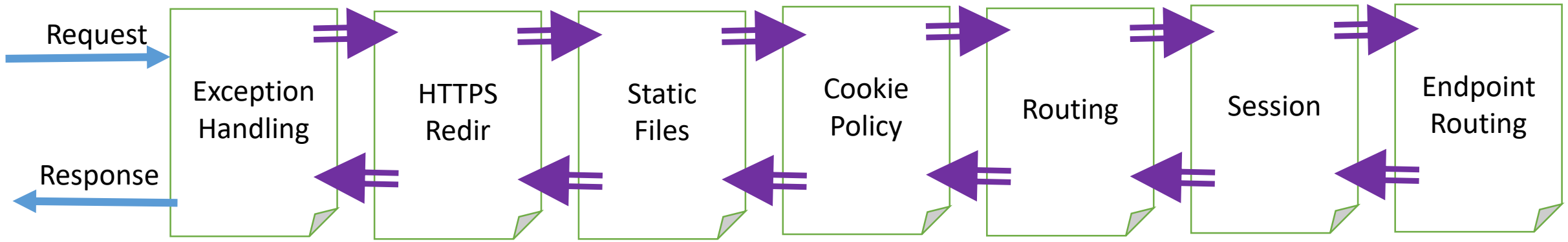
Updated https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

The Middleware

- It is software service provided to software applications which serves as a glue between different components with i/o communication channels that serves a specific purpose.
- The middleware provides the essential building blocks how an application handles HTTP requests.
- A series of components that form the application request pipeline.
- The middleware provides all kinds of foundational application-level infrastructural components for .NET Core application development.
- Examples of middleware are: routing, session, CORS, authentication, caching

Middleware Pipeline

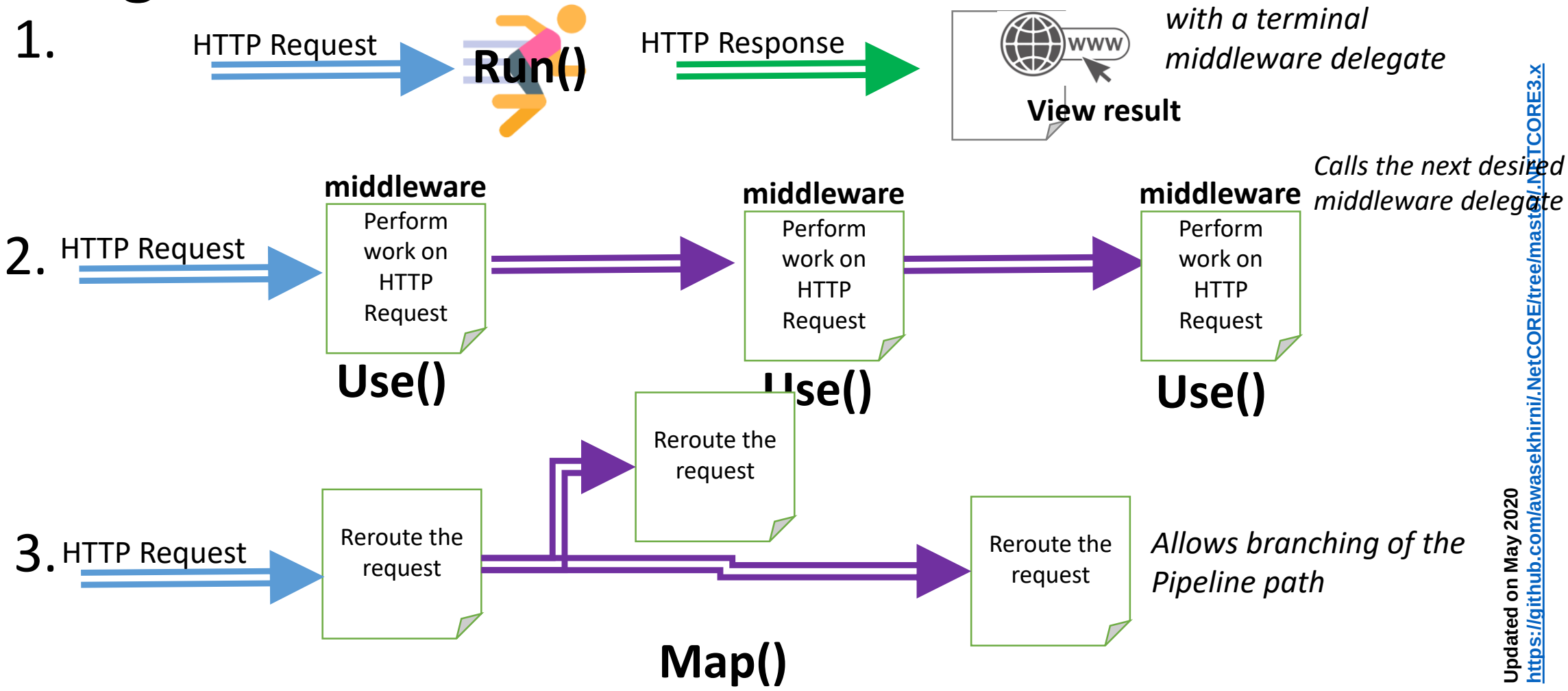
A typical order of middleware layers in an ASP.NET Core web application



Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

Registering Middleware Components using Configuration Options



Middleware Configuration Options

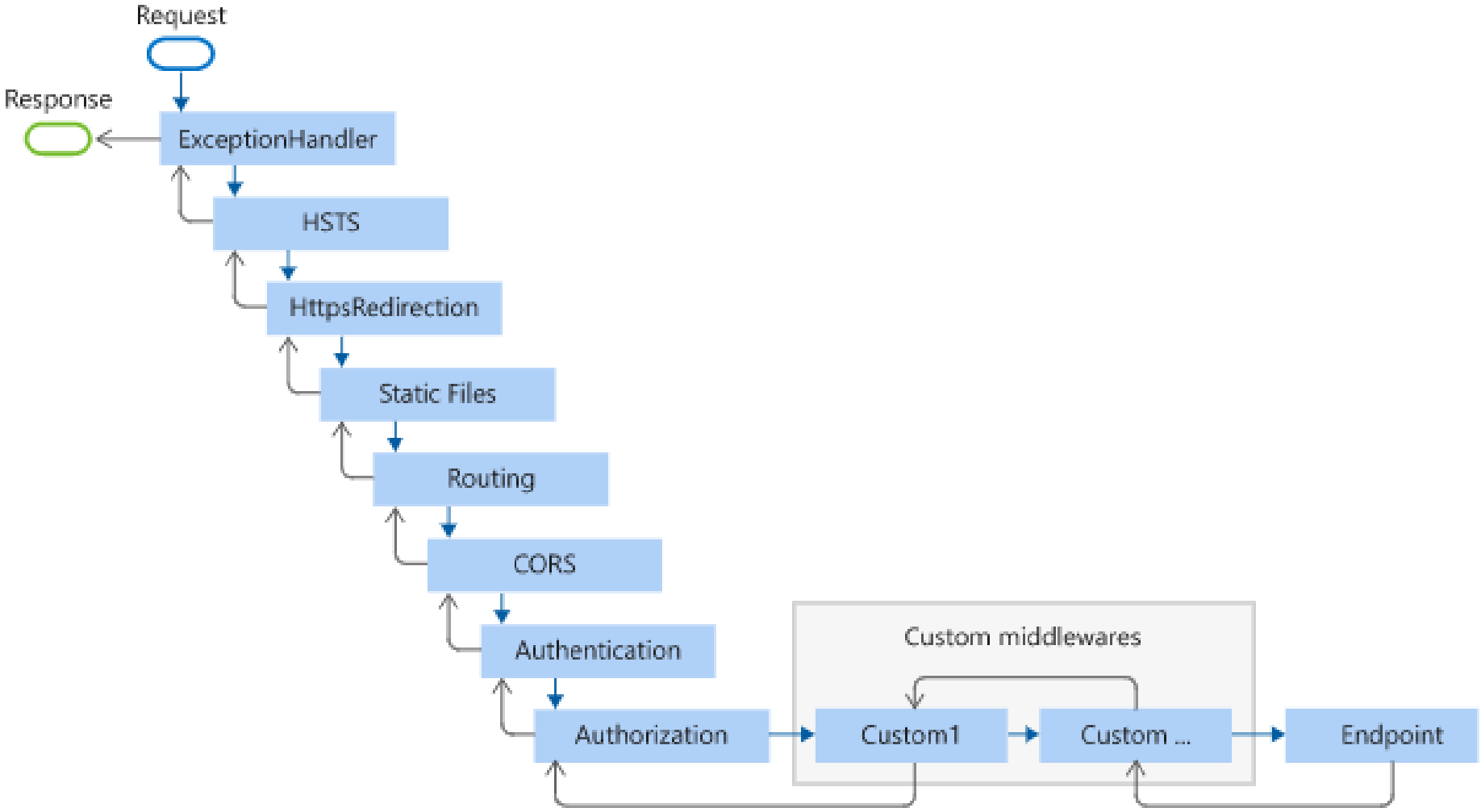
Option	Description	Usage

Run	It terminates the middleware pipeline to render view result.	app.Run(async context =>{
		await context.Response.WriteAsync("Render to Webpage")
		}

Use	It forwards the request onto the **next** delegate	app.Use(async (context,next) =>{
		await next.Invoke();
		}

Map	It redirect incoming requests which end in "search" to execute the **GlobalSearch** method.	app.Map("/search", GlobalSearch);
		
		Private static void GlobalSearch (IApplicationBuilder app){
		app.Run(async context=>{
		Await context.Response.WriteAsync("search results here")
		});
		}

Middleware Order



Built-in Middleware

Middleware	Description
Authentication	Adds authentication support
CORS	Configures Cross-Origin Resource Sharing
Routing	Adds routing capabilities for MVC or web forms
Session	Adds support for user session
StaticFiles	Adds support for serving static files and directory browsing
Diagnostics	Adds support for reporting and handling exceptions and errors

Middleware Methods

Exception Handling	UseDeveloperExceptionPage()	Used in development to catch run-time exceptions
	UseDatabaseErrorPage()	
	UseExceptionHandler()	Used in production for run-time exceptions
HSTS &HTTPS Redirection	UseHsts()	Used in production to enabled HSTS(HTTP Strict Transport Security protocol) and enforces HTTPS
	UseHttpsRedirection()	Forces HTTP calls to automatically redirect to equivalent HTTPS addresses.
Static Files	UseStaticFiles()	Used to enable static files such as HTML, javascript, CSS and graphics files called early to avoid the need for authentication, session or MVC middleware
Cookie Policy	UseCookiePolicy()	Used to enforce cookie policy and display GDPR-friendly messaging
Authentication, Authorization and Sessions	UseAuthentication()	Used to enable authentication and then subsequently allow authorization
	UseSesssion()	Manually added to the Startup file to enable the Session middleware.

Middleware Methods

Endpoint Routing	UseEndpoints()	Usage varies on project templates used for MVC, Razor Pages and Blazor
	MapControllerRoute()	Set the default route and any custom routes when using MVC
	BlazorHub()	Sets up Blazor Hub

Diagnostics Middleware

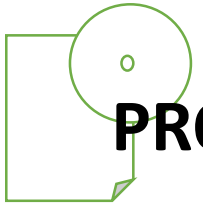
Middleware	Extension Method	Description
DeveloperExceptionPageMiddleware	UseDeveloperExceptionPage()	Captures synchronous and asynchronous exceptions fromt eh pipeline and generates HTML error responses
ExceptionHandlerMiddleware	UseExceptionHandler()	Catch exceptions, log them and re-execute in an alternate pipeline
StatusCodePagesMiddleware	UseStatusCodePages()	Check for response with status codes between 400 and 599
WelcomePageMiddleware	UseWelcomePage()	Display Welcome page for the root path.

Custom Middleware

- Create a custom middleware class
- A constructor accepting the “next” middleware delegate.
- Invoke method to execute logic
- Perform work on the request before forwarding or generating a response.

```
public class CustomMiddleware{

private Request Delegate _next;

public CustomMiddleware(RequestDelegate next) {
 _next = next;
 }

 public Task Invoke(HttpContext context){

 await _next.Invoke(context)
 }

 }
```

When/Where Middleware Components Execute



PROGRAM CLASS

- Program class is used as the entry point to start the entire application

STARTUP CLASS

- It is used for configuring the middleware components.

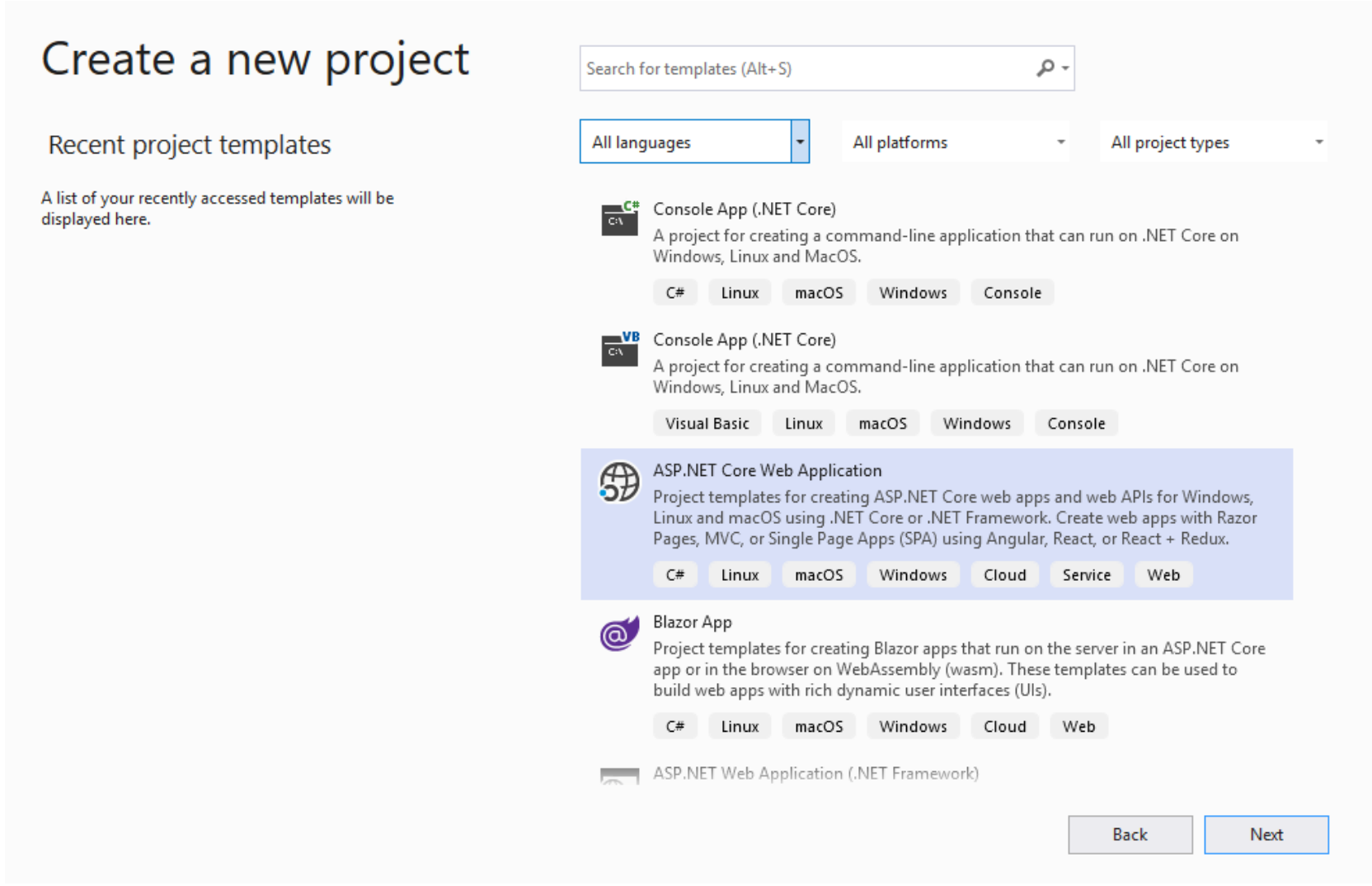
SYED AWASE KHIRNI

Example 8: ASP.NET Core Custom Middleware Demo

Learning Outcomes:

1. **Learning Outcome 1**
2. **Learning Outcome 2**

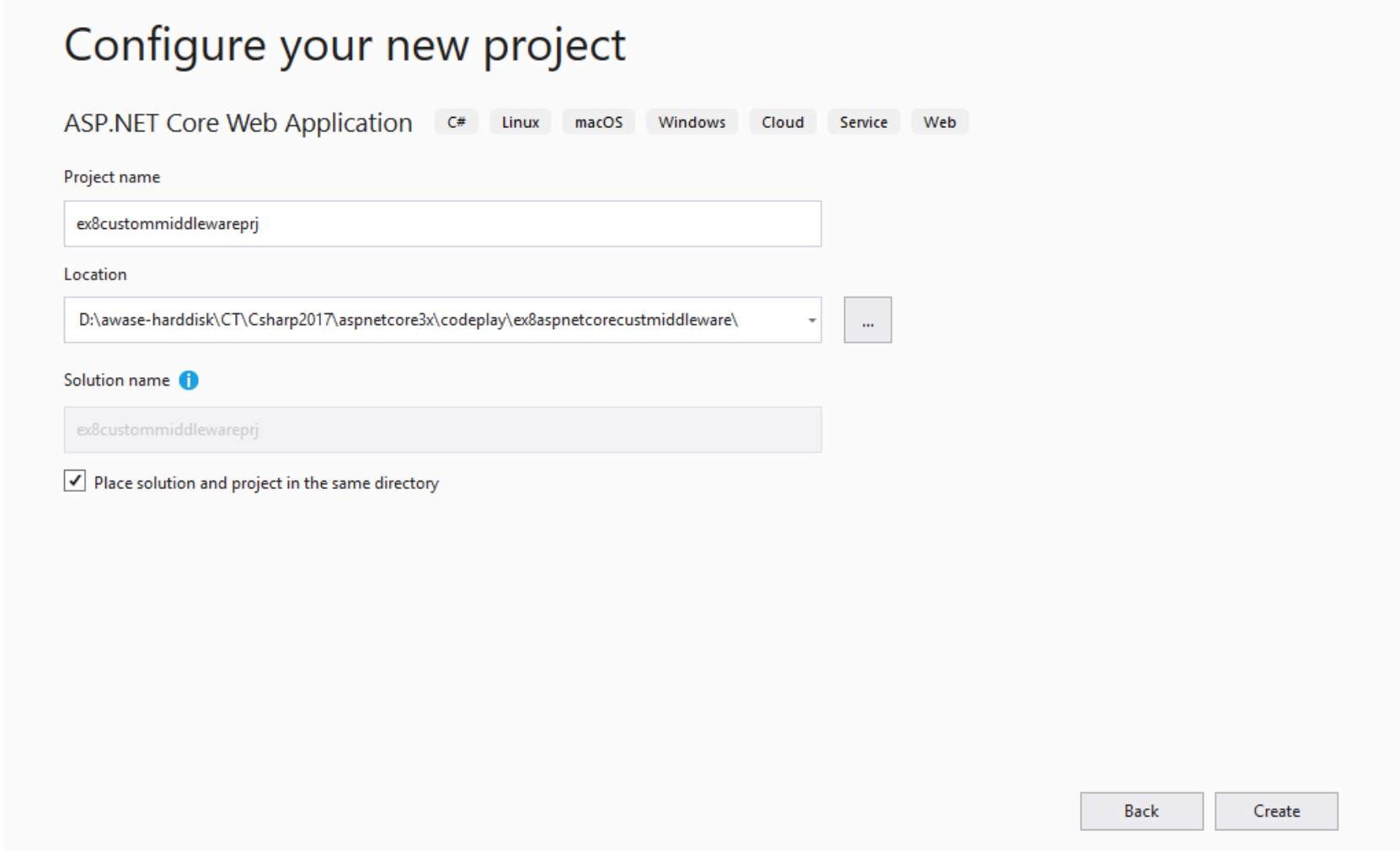
Example: 8



Original Series

Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

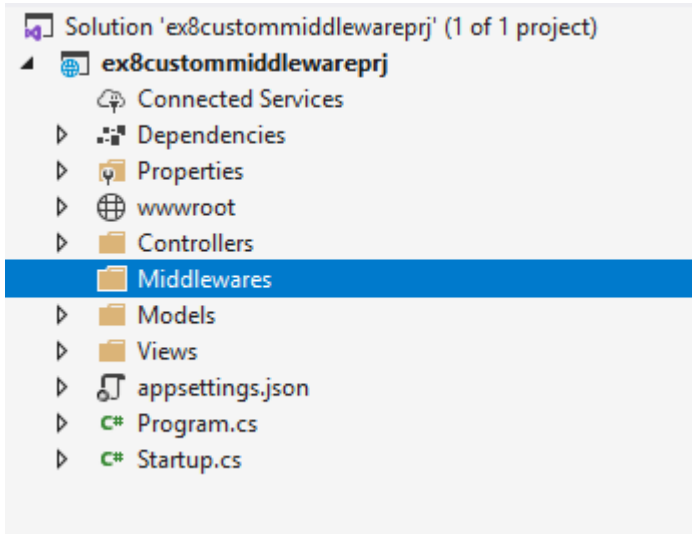
Example: 8

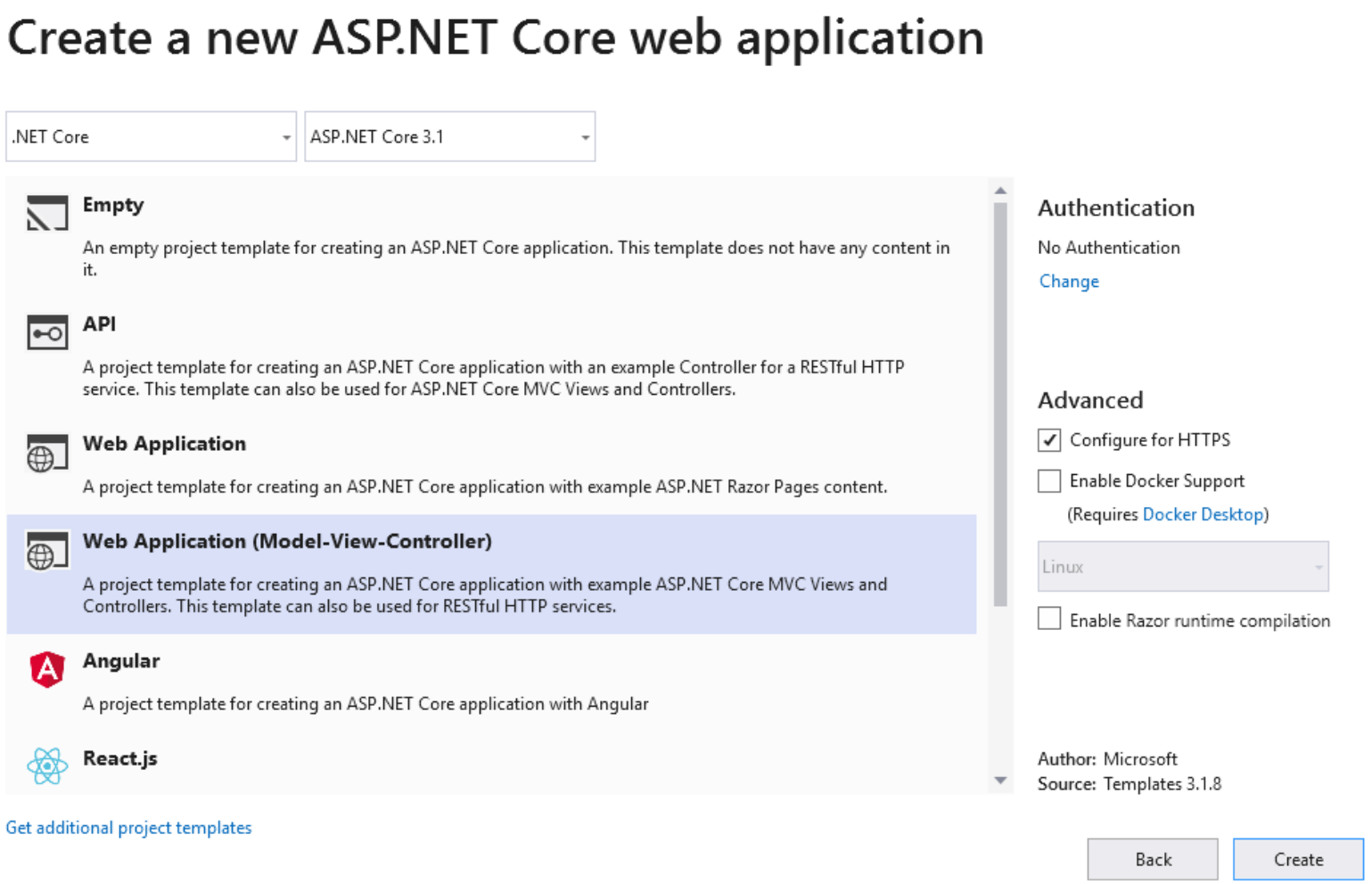


Original Series

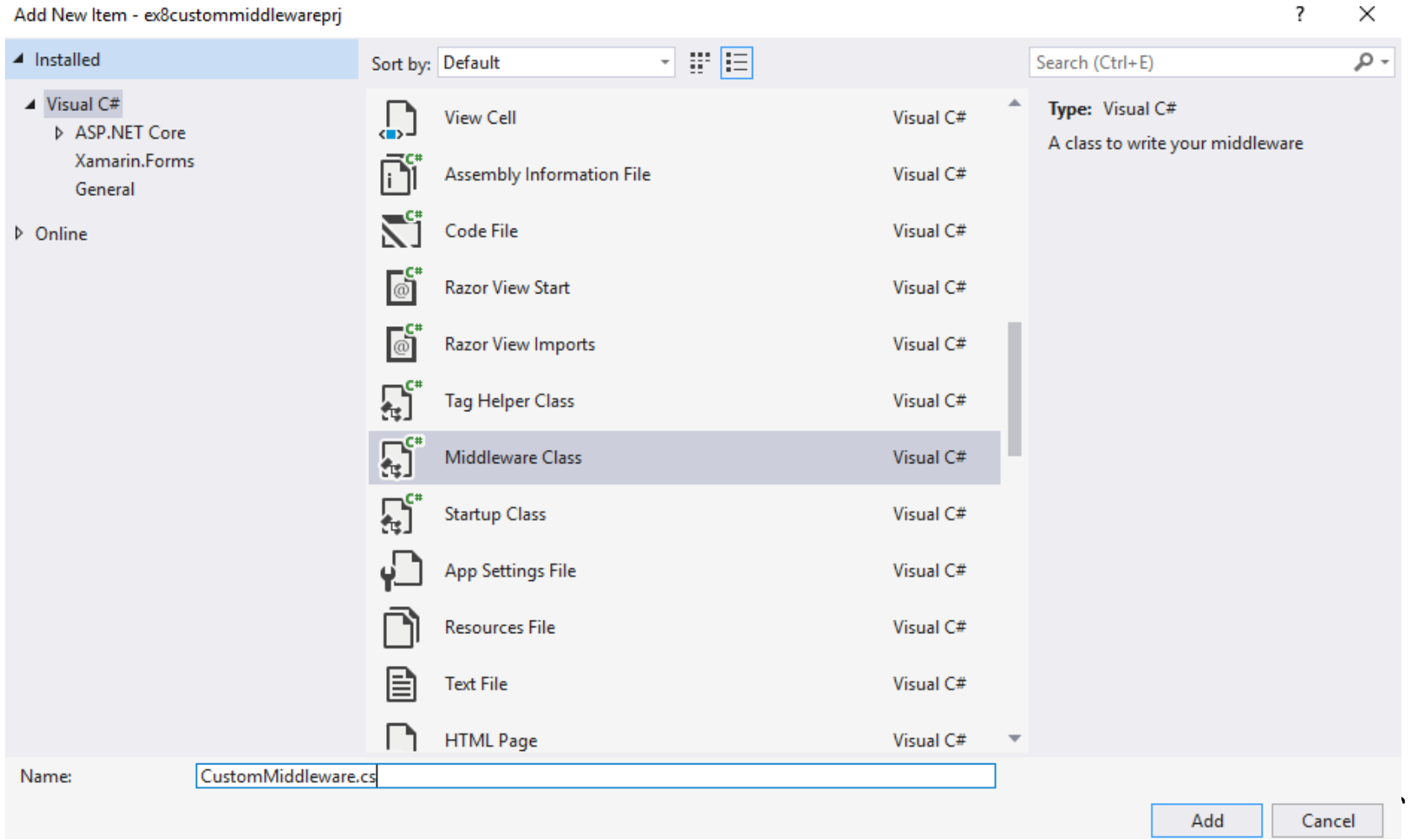
Updated on May 2020
https://github.com/awasekhirni/.NetCORE/tree/master/.NETCORE3.x

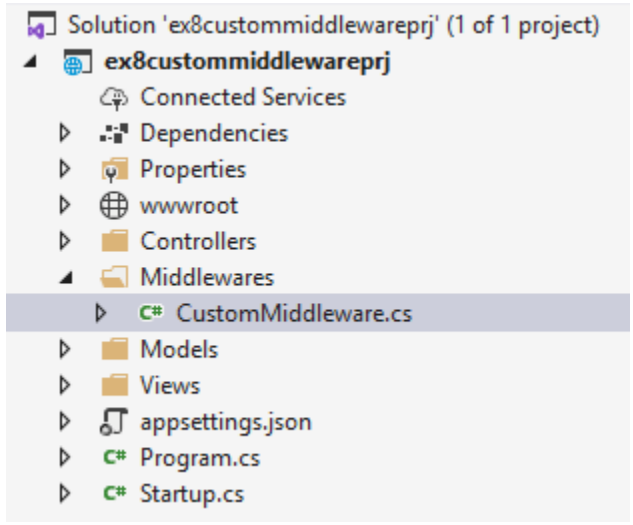
- Create ***Middlewares*** *folder*

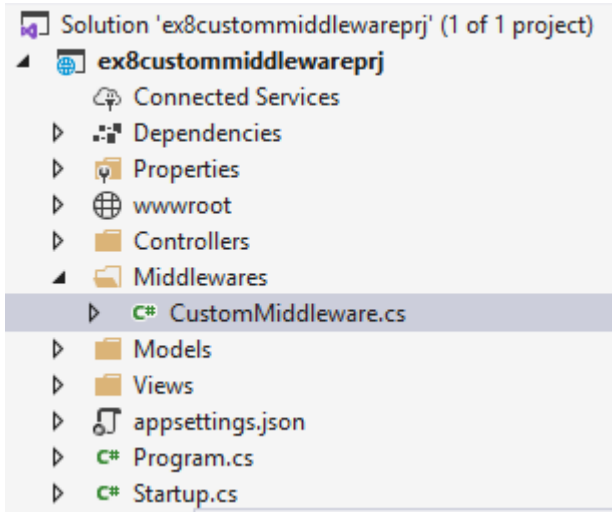




Example: 8







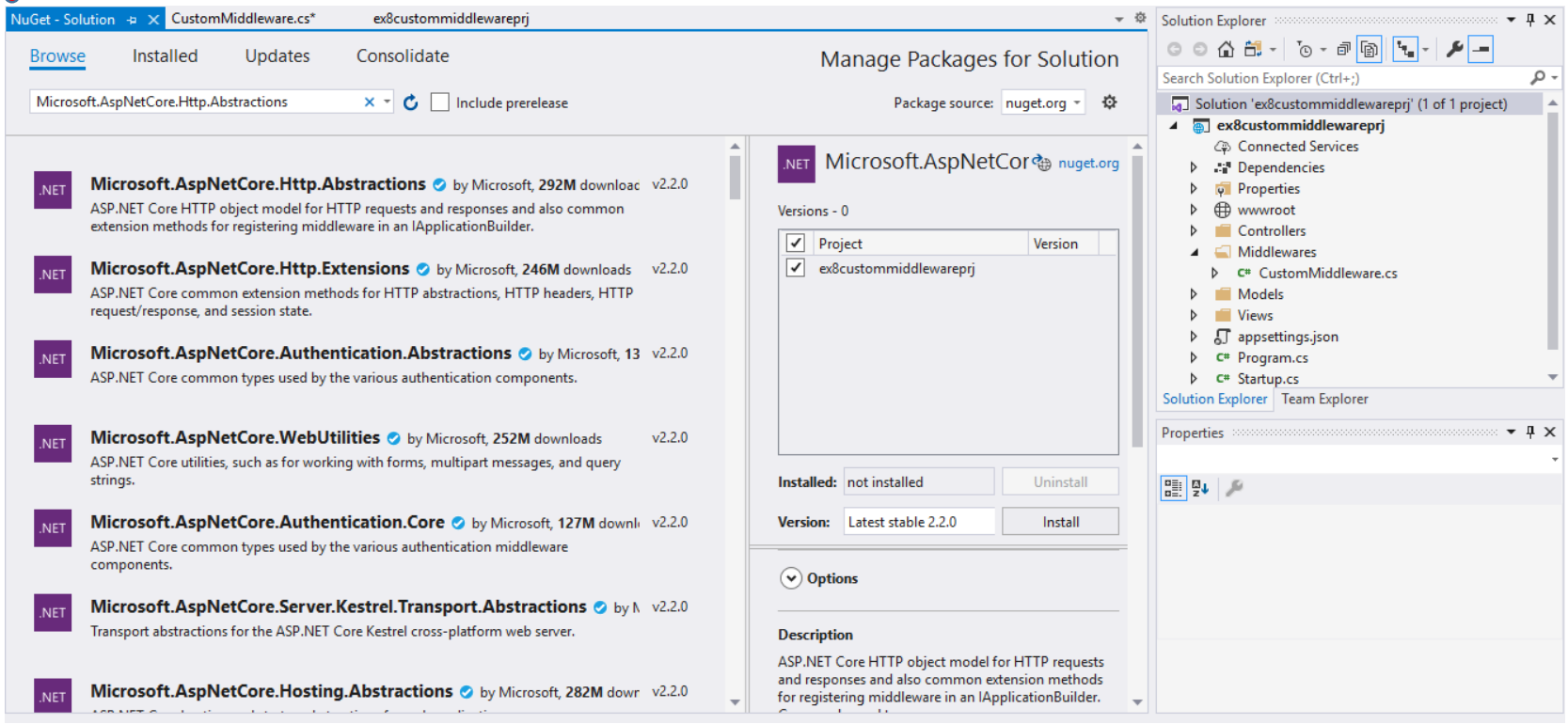

```
public class CustomMiddleware
{
    private readonly RequestDelegate _next;
    private readonly ILogger _logger;

    0 references
    public CustomMiddleware(RequestDelegate next, ILoggerFactory logFactory)
    {
        _next = next;
        _logger = logFactory.CreateLogger("CustomMiddlewareLog");
    }

    0 references
    public async Task Invoke(HttpContext httpContext)
    {
        _logger.LogInformation("Syed Awase Khirni--Custom MiddlewareLog is Executing Now---");
        await _next(httpContext); //calling the next middleware in the pipeline
    }
}
```

- Install "***Microsoft.AspNetCore.Http.Abstractions***" *to the solutions*

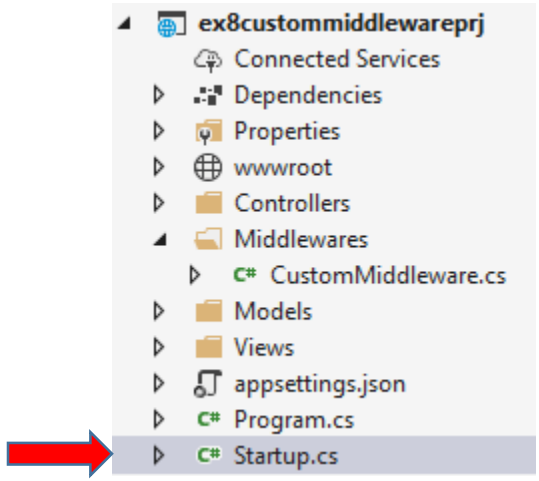
Check for Installation of Dependency libraries for 3.x

Microsoft.AspNetCore.Http.Abstractions

Else install

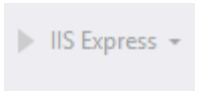


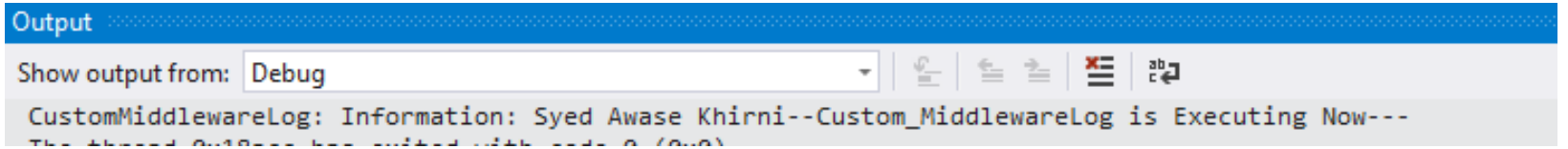
Example: 8



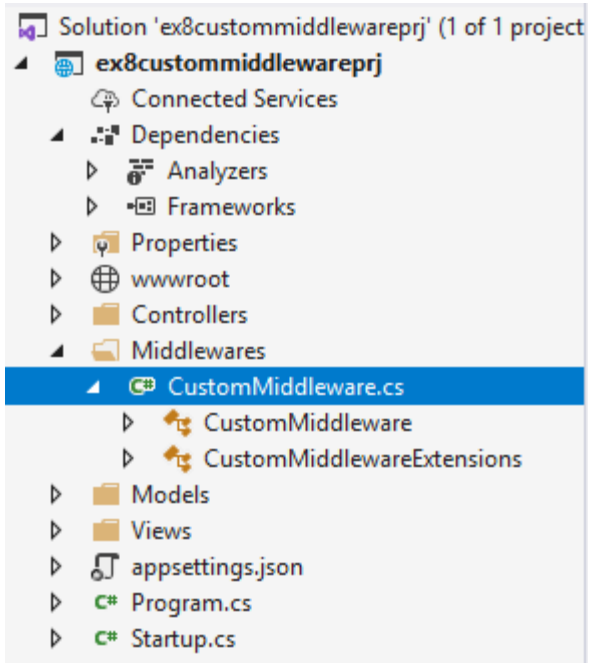

```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
0 references
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{

    //adding the custom middleware here
    app.UseCustomMiddleware();
```

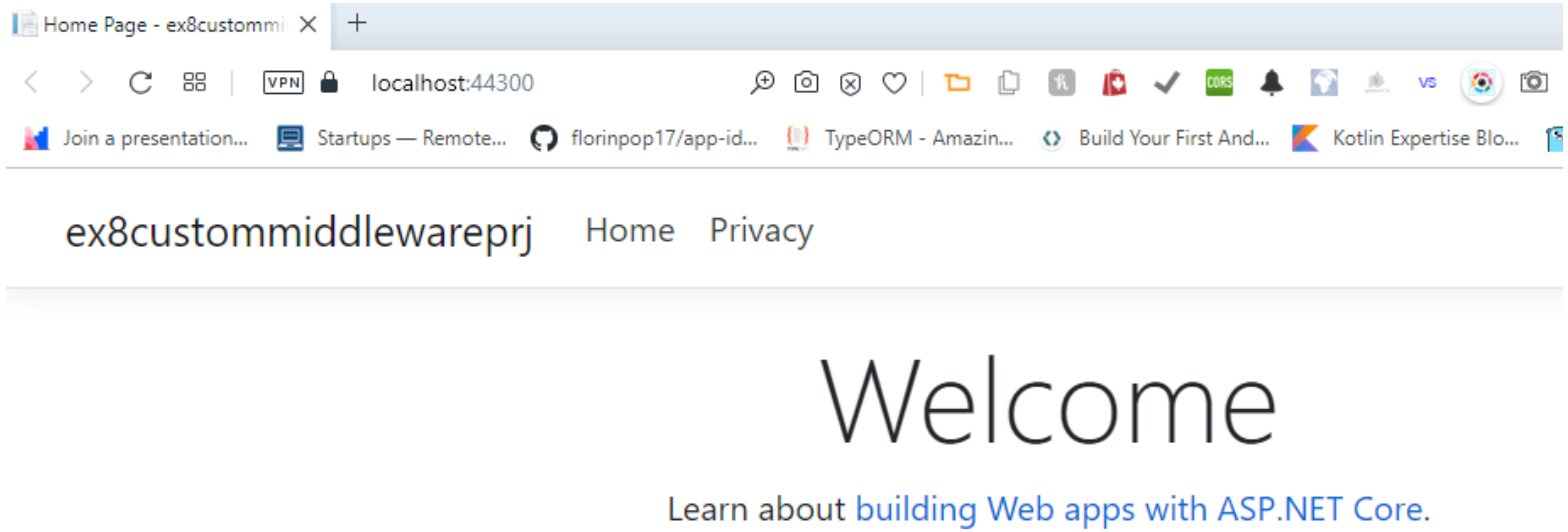
Example: 8





- Lets build our custom middleware





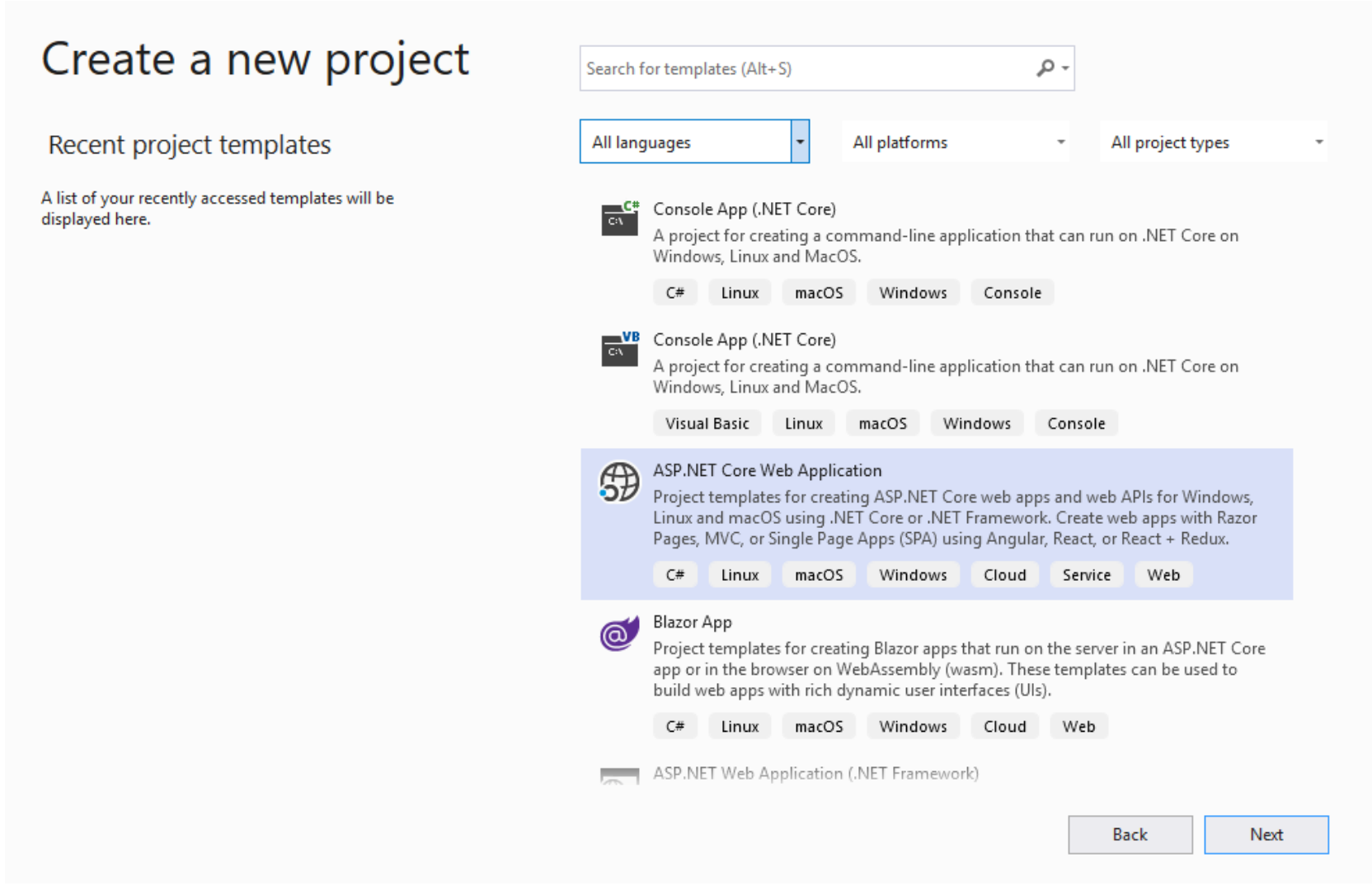
SYED AWASE KHIRNI

Example 9: ASP.NET Core 3 Online Users Middleware

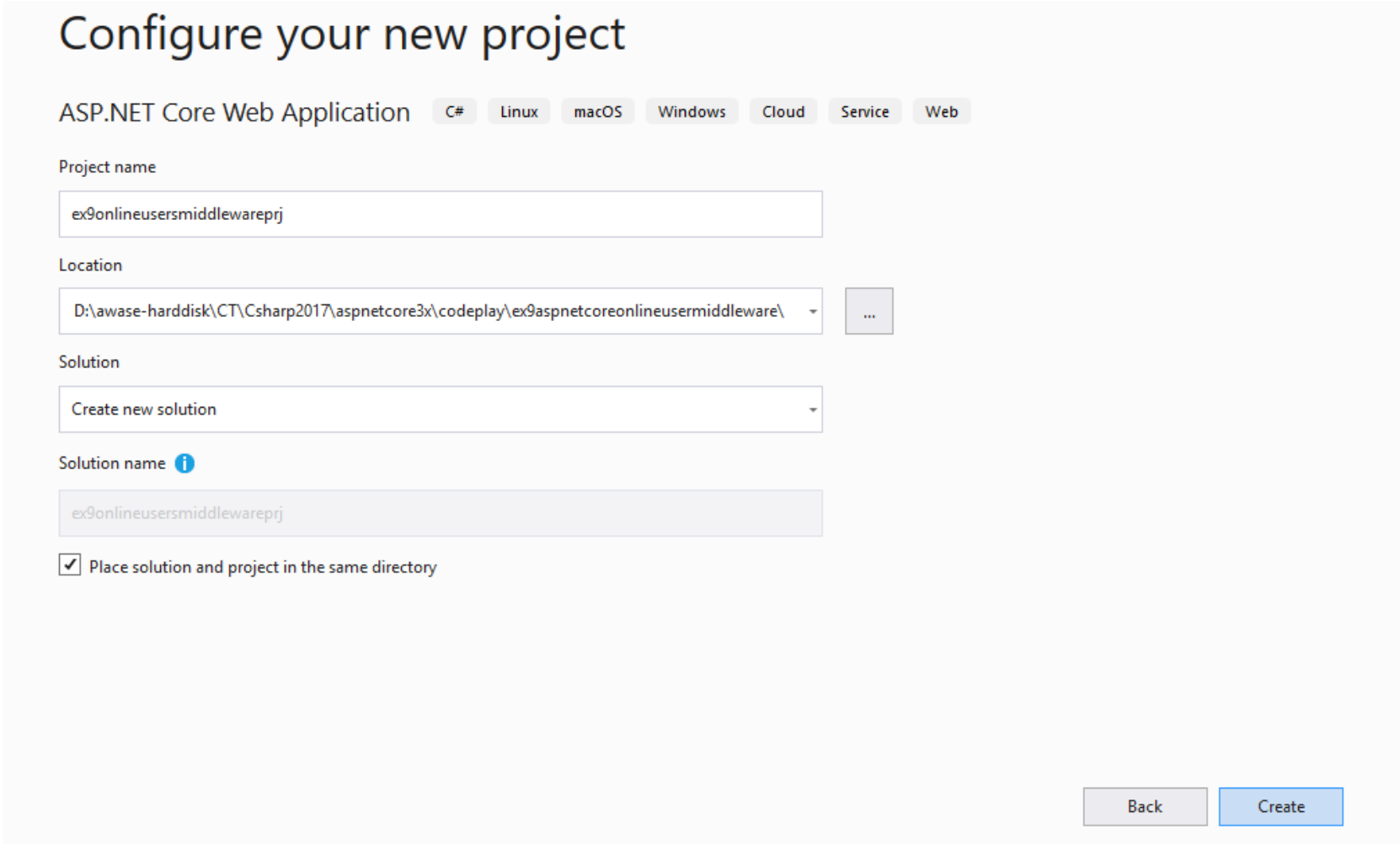
Learning Outcomes:

1. **Learning Outcome 1**
2. **Learning Outcome 2**

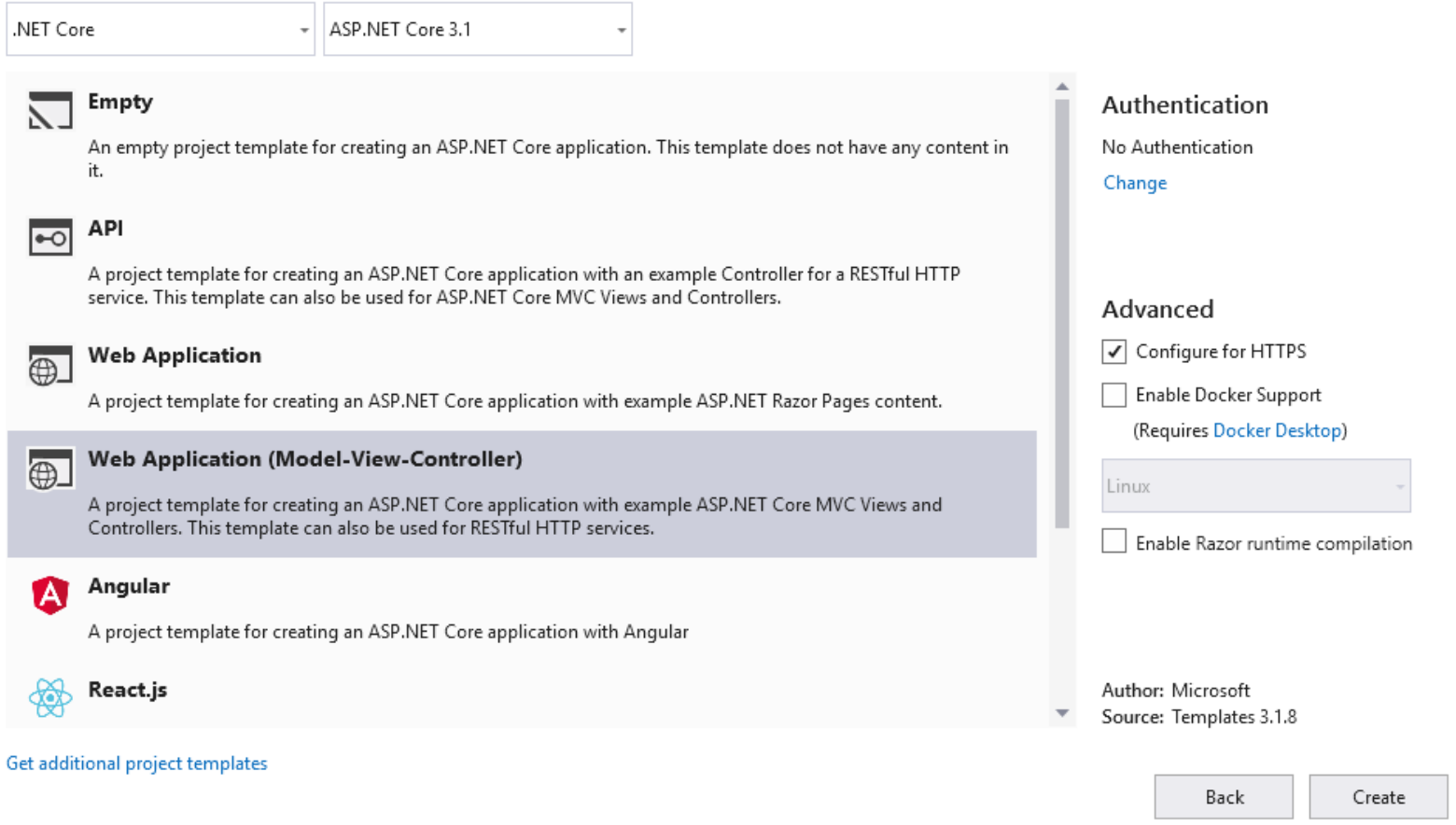
Example: 9



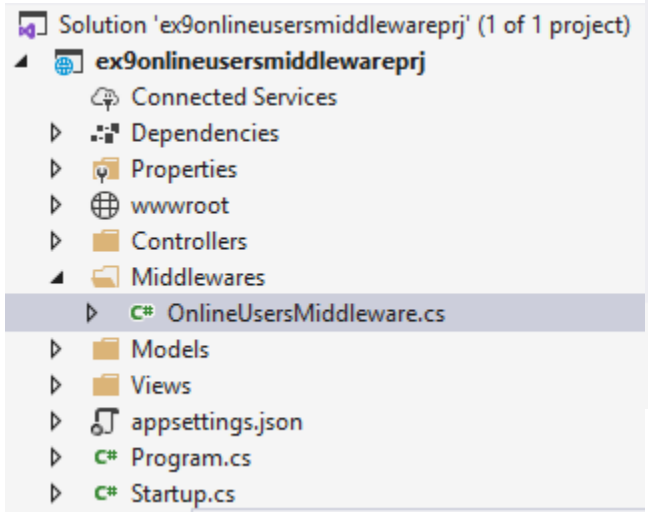
Original Series

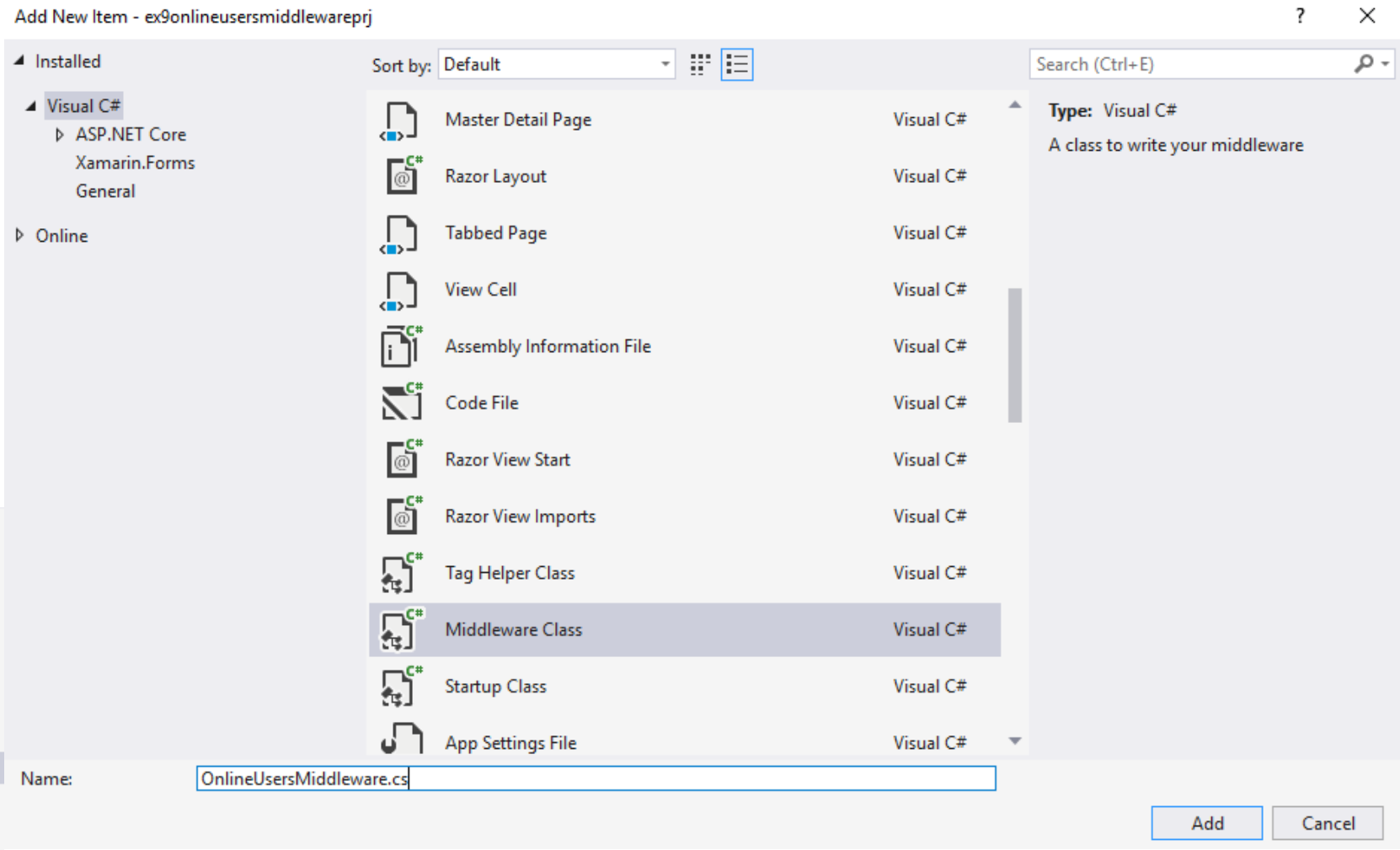


Create a new ASP.NET Core web application



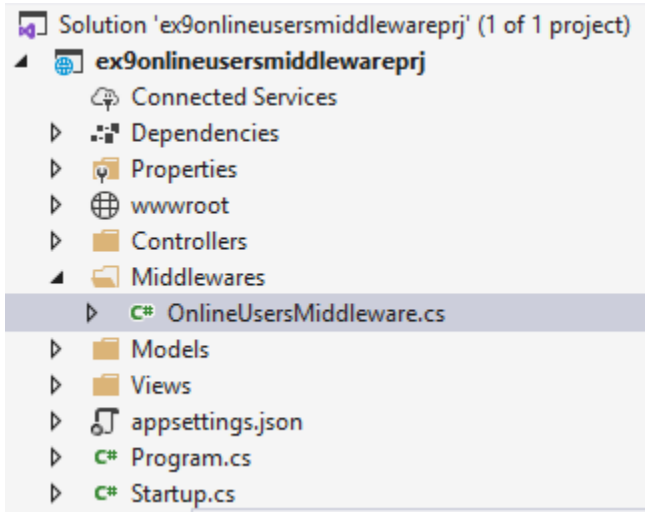
Example: 9



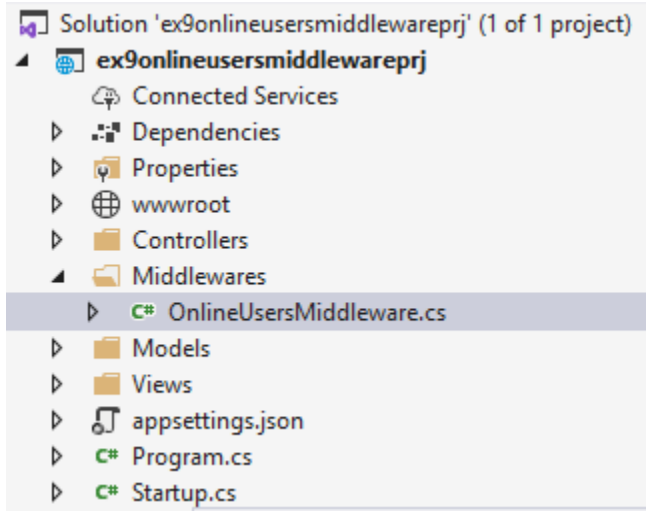


OnlineUsersMiddleware.cs


```
namespace ex9onlineusersmiddlewareprj.Middlewares
{
    // You may need to install the Microsoft.AspNetCore.Http.Abstractions package into your project
    3 references
    public class OnlineUsersMiddleware
    {
        private readonly RequestDelegate _next;

        private readonly string _cookieName;
        private readonly int _lastActivityMinutes = 20;

        private static readonly ConcurrentDictionary<string, bool> _allKeys = new ConcurrentDictionary<string, bool>();

        0 references
        public OnlineUsersMiddleware(RequestDelegate next, string cookieName ="UserGuid", int lastActivityMinutes=20)
        {
            _next = next;
            _cookieName = cookieName;
            _lastActivityMinutes = lastActivityMinutes;
        }
```

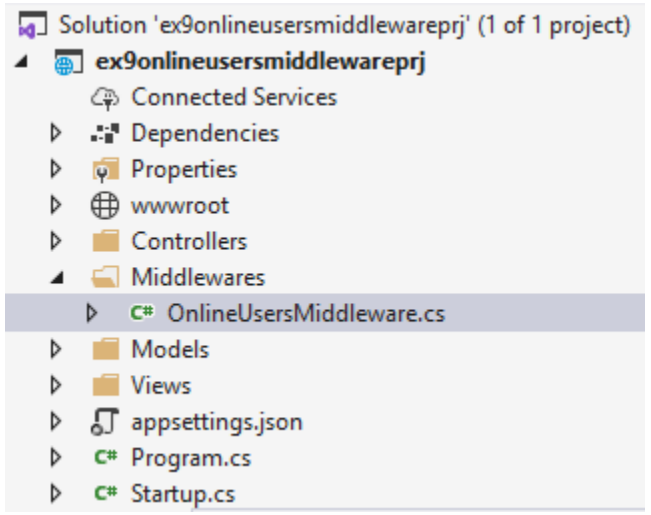


OnlineUsersMiddleware.cs



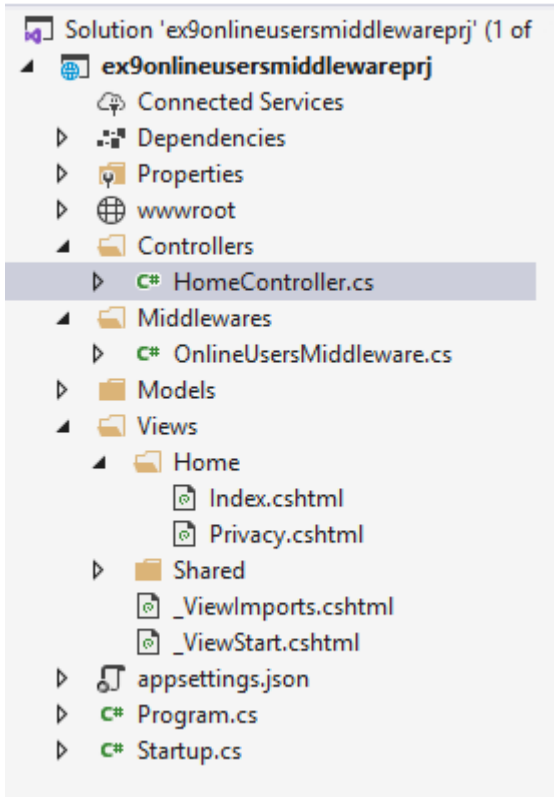

```
0 references
public Task InvokeAsync(HttpContext httpContext, IMemoryCache memoryCache)
{
    if (httpContext.Request.Cookies.TryGetValue(
        _cookieName,
        out var userGuid) == false)
    {
        userGuid = Guid.NewGuid().ToString();
        httpContext.Response.Cookies.Append(_cookieName, userGuid,
            new CookieOptions { HttpOnly = true, MaxAge = TimeSpan.FromDays(10) }
            );
    }
    memoryCache.GetOrCreate(userGuid, cacheEntry => {

        if(_allKeys.TryAdd(userGuid, true) == false)
        {
            //if add key failed, set the expiration to the past
            cacheEntry.AbsoluteExpiration = DateTimeOffset.MinValue;
        }
        else
        {
            cacheEntry.SlidingExpiration = TimeSpan.FromSeconds(10);
            cacheEntry.RegisterPostEvictionCallback(RemoveKeyWhenExpired);
        }
        return string.Empty;
    });

    return _next(httpContext);
}
```

OnlineUsersMiddleware.cs



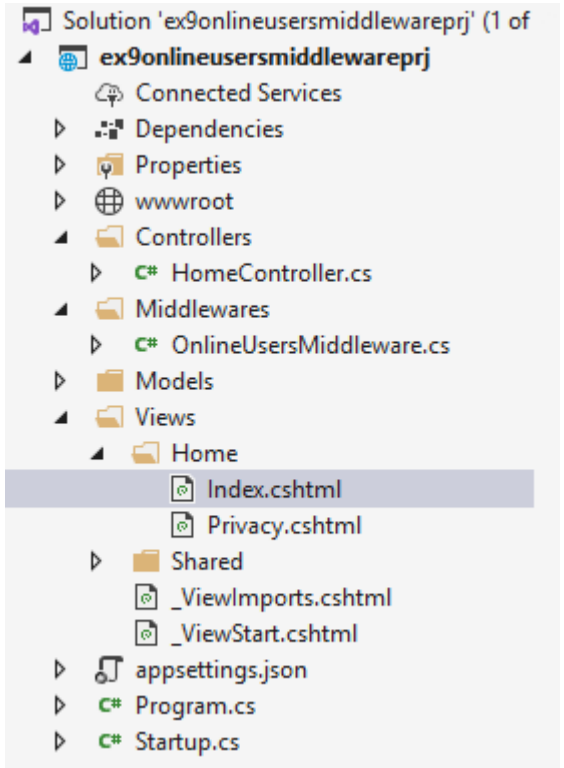

```
        //step4- add the method to remove the key when expired
        1 reference
        private void RemoveKeyWhenExpired(object key, object value, EvictionReason reason, object state)
        {
            string strKey = (string)key;
            if (!_allKeys.TryRemove(strKey, out _))
                _allKeys.TryUpdate(strKey, false, true);
        }


        //step 5- get the no of online users count to render on the view pass this value to the controller
        1 reference
        public static int GetOnlineUsersCount() => _allKeys.Count(p => p.Value);
    }

    // Extension method used to add the middleware to the HTTP request pipeline.
    0 references
    public static class OnlineUsersMiddlewareExtensions
    {
        1 reference
        public static IApplicationBuilder UseOnlineUsersMiddleware(this IApplicationBuilder builder, string
          cookieName="UserGuid", int lastActivityMinutes=20)
        {
            return builder.UseMiddleware<OnlineUsersMiddleware>(cookieName,lastActivityMinutes);
        }
    }
```

HomeController.cs




```
0 references
public IActionResult Index()
{
    //step 6- modelbinding the count of users to render to the view model
    var onlineUsersCount = OnlineUsersMiddleware.GetOnlineUsersCount();
    return View(onlineUsersCount);
}
```

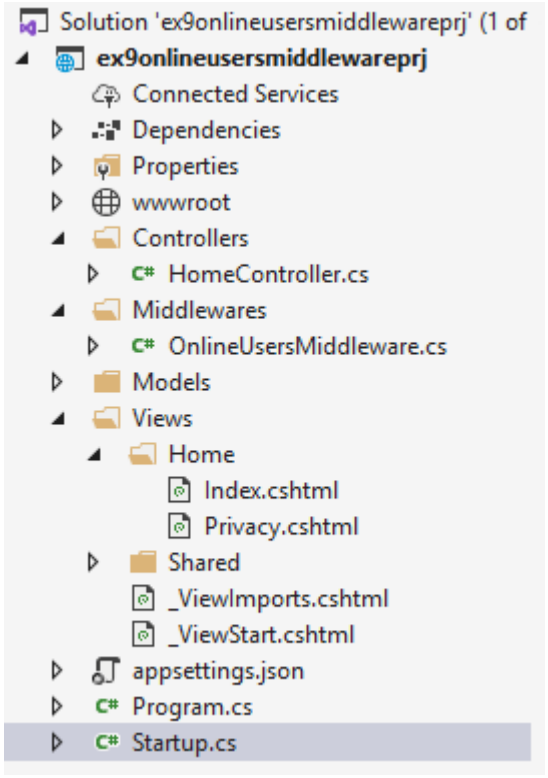
Views/Home/Index.cshtml



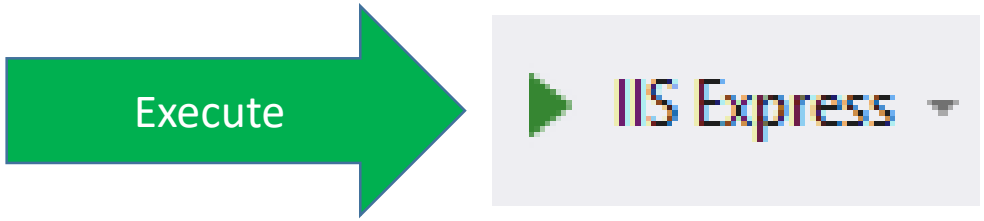

```
@{
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <h1 class="display-4">Welcome</h1>
    <p>Learn about <a href="https://docs.microsoft.com/aspnet/core">building Web apps with ASP.NET Core</a>.</p>
</div>

<!--step 7- pass the online users here modelbinding happens here -->
<div class="text-center">
    <h1 class="display-2">Online Users:@Model</h1>
    <p> Displaying the number of online users using middleware</p>
</div>
```

Example: 9

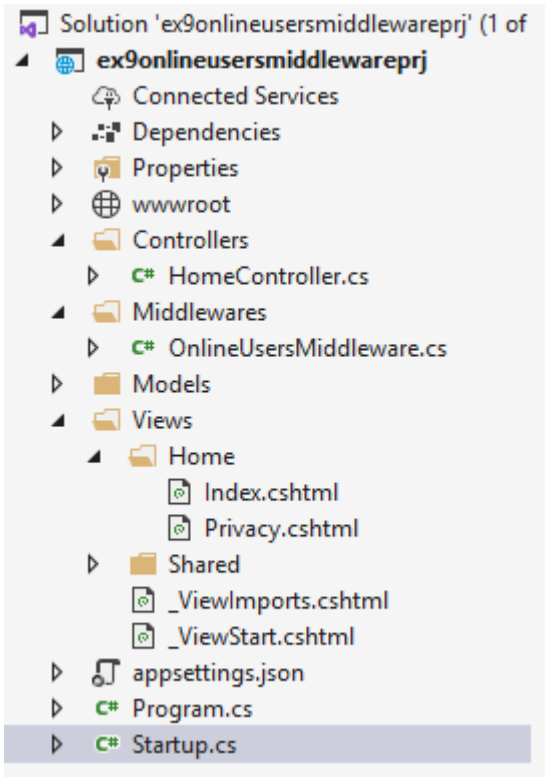
Startup.cs

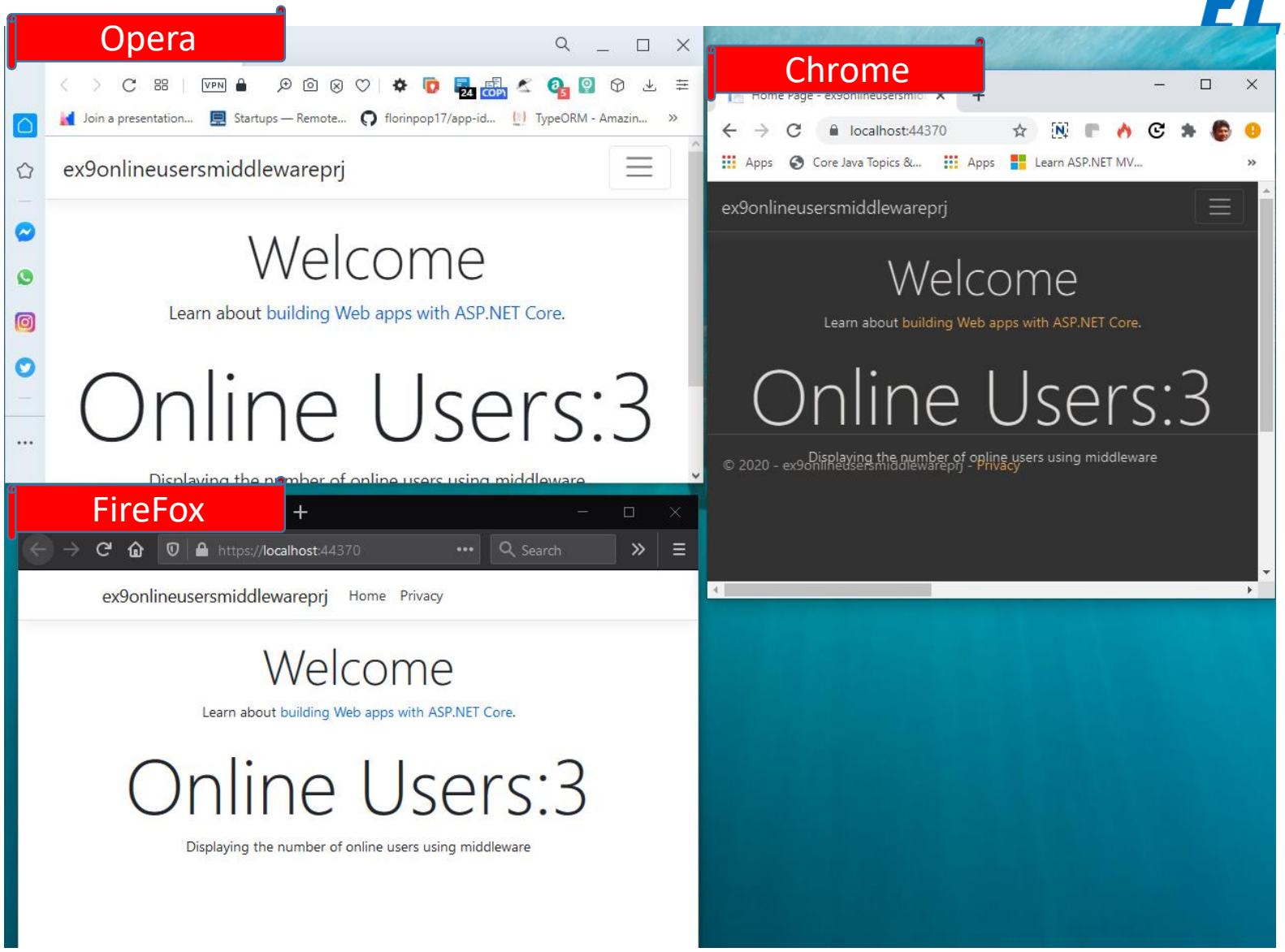



```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
0 references
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    //step8- register the custom middleware method to call in the request pipeline
    //adding the custom middleware to the configure method
    app.UseOnlineUsersMiddleware();
```



Example: 9





https://vicluar.wordpress.com/2020/06/01/implement-health-check-in-asp-net-core-services/

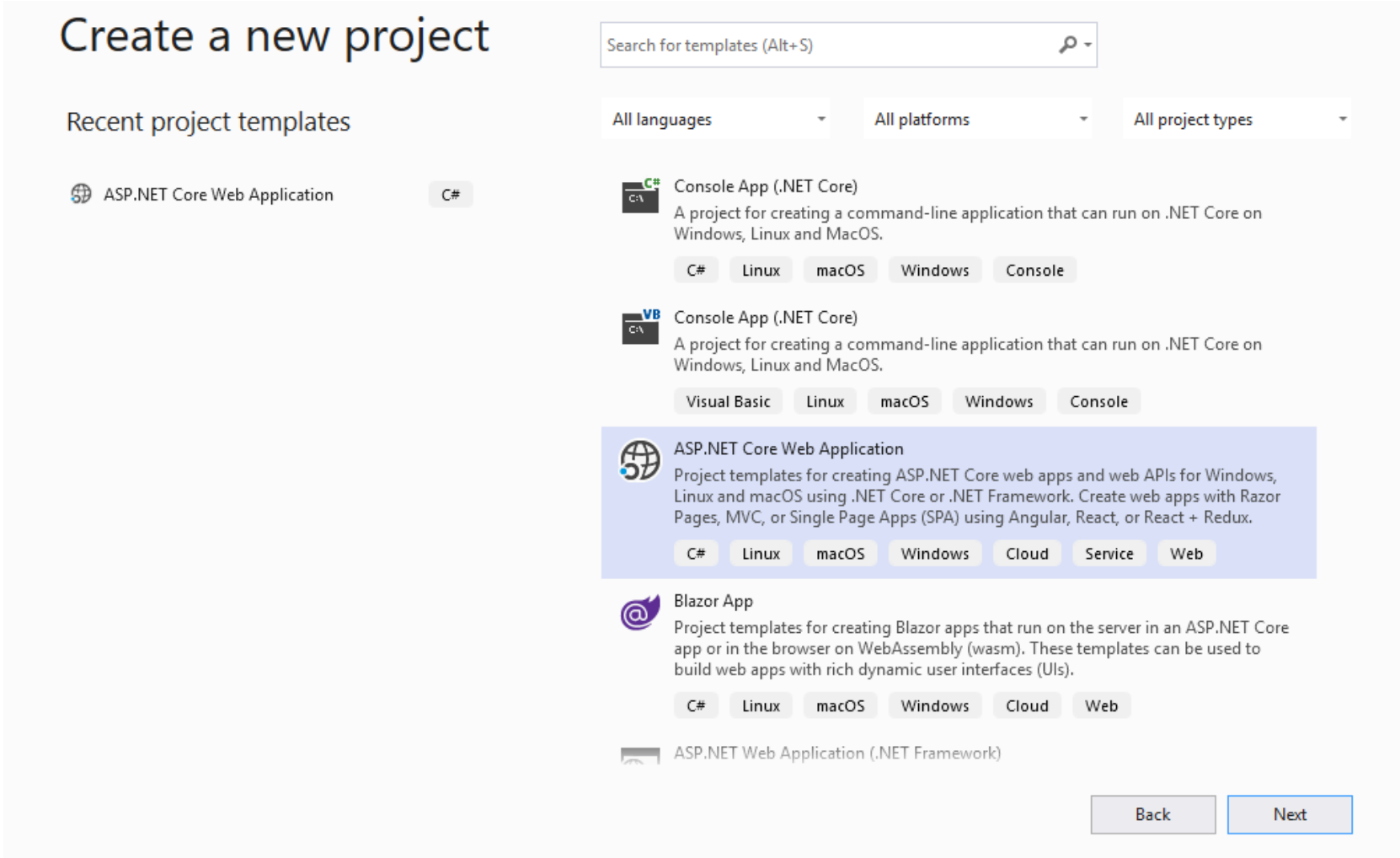
SYED AWASE KHIRNI

Example 10 : ASP.NET Core Health Check Services

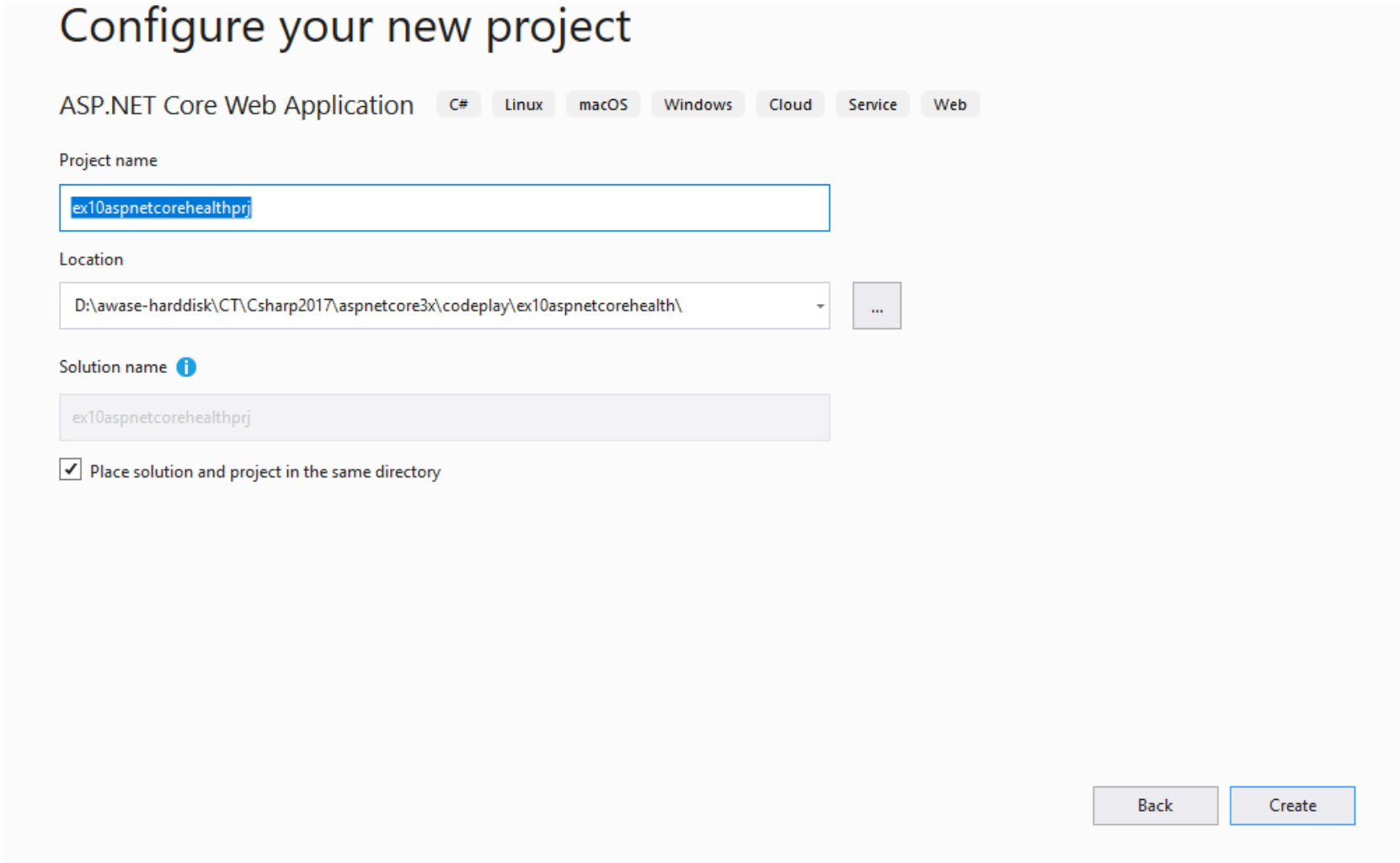
Learning Outcomes:

1. **Learning Outcome 1**
2. **Learning Outcome 2**

Example: 10



Example:



Example:

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

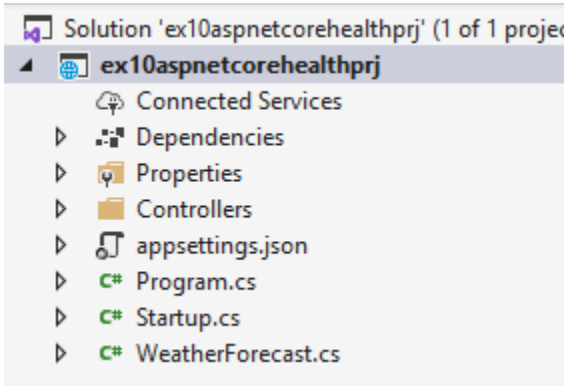
Get additional project templates

Authentication
No Authentication
Change

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

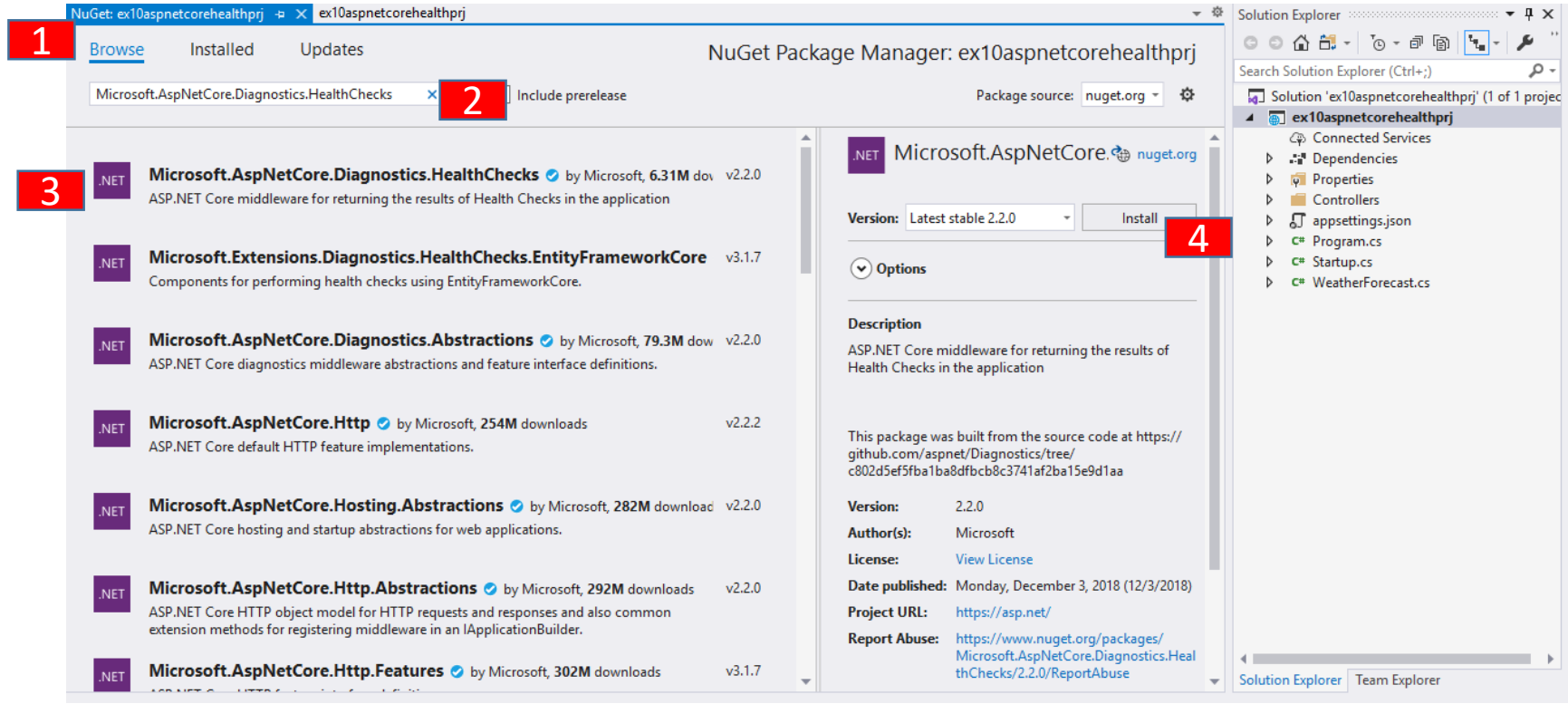
Author: Microsoft
Source: Templates 3.1.8

Back | Create

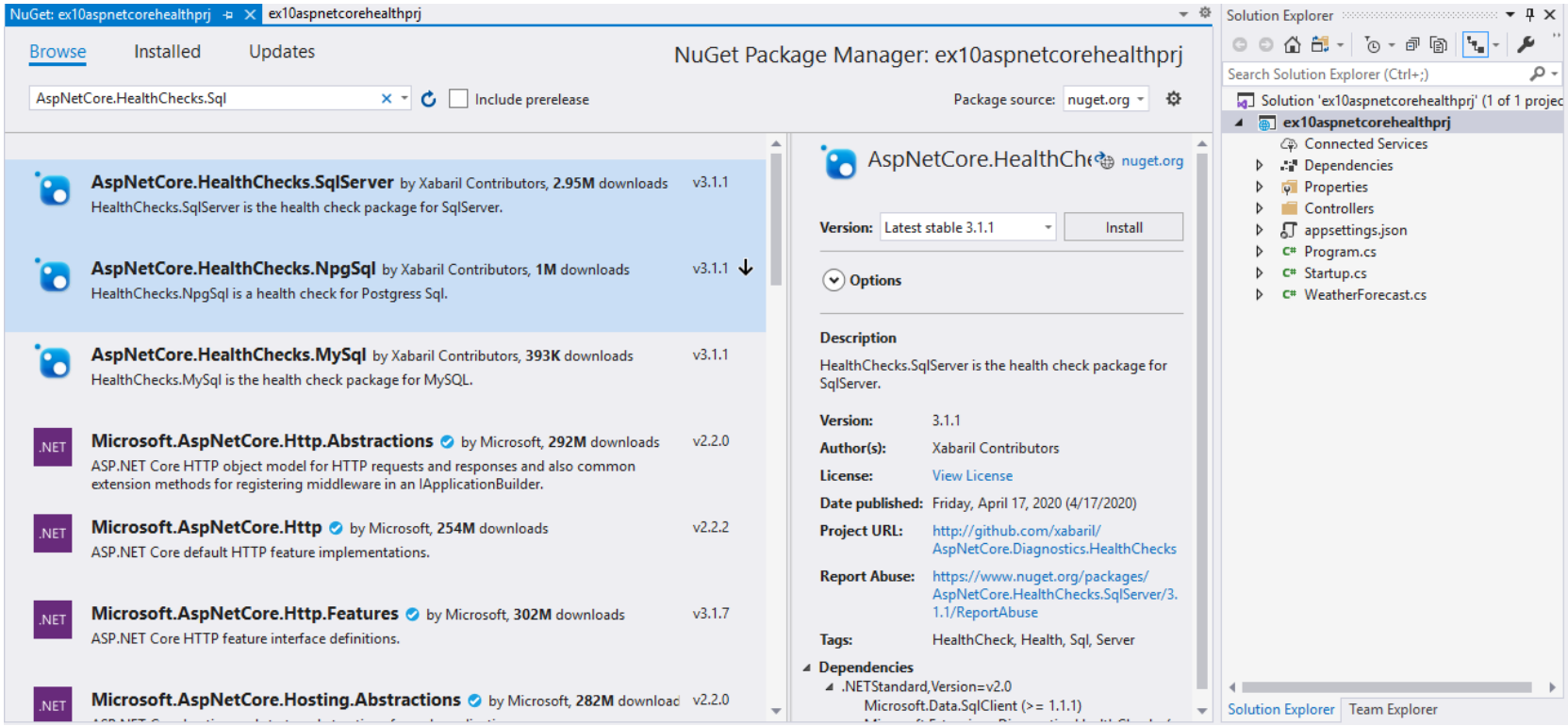


Example:

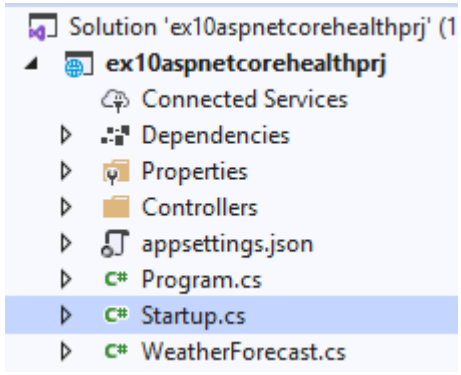
Microsoft.AspNetCore.Diagnostics.HealthChecks



AspNetCore.HealthChecks.SqlServer



Example:



Startup.cs

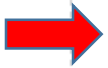
ex10aspnetcorehealthprj | ex10aspnetcorehealthprj.Startup | ConfigureServices(IServiceCollection services)

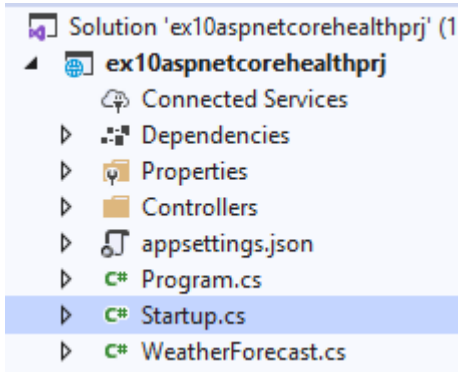
```
0 references
public void ConfigureServices(IServiceCollection services)
{
    services.AddControllers();
    //step 1: add services healthchecks
    services.AddHealthChecks()
        .AddSqlServer(
        $"Server=localhost\\SQLEXPRESS;" +
        $"Database=ExampleTen;" +
        $"Trusted_Connection=True;" +
        $"MultipleActiveResultSets=true;" +
        $"ConnectRetryCount=0"
        );

}
```

Example:

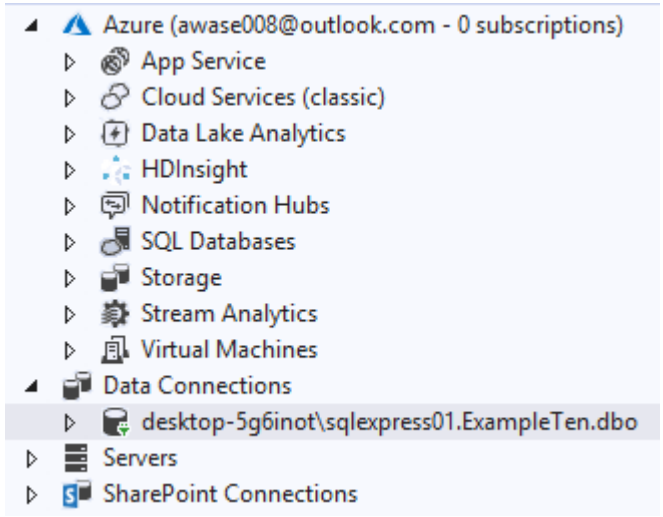
Startup.cs | ex10aspnetcorehealthprj | ex10aspnetcorehealthprj.Startup | Configure(IApplicationBuilder app, IWebHostEnvironr

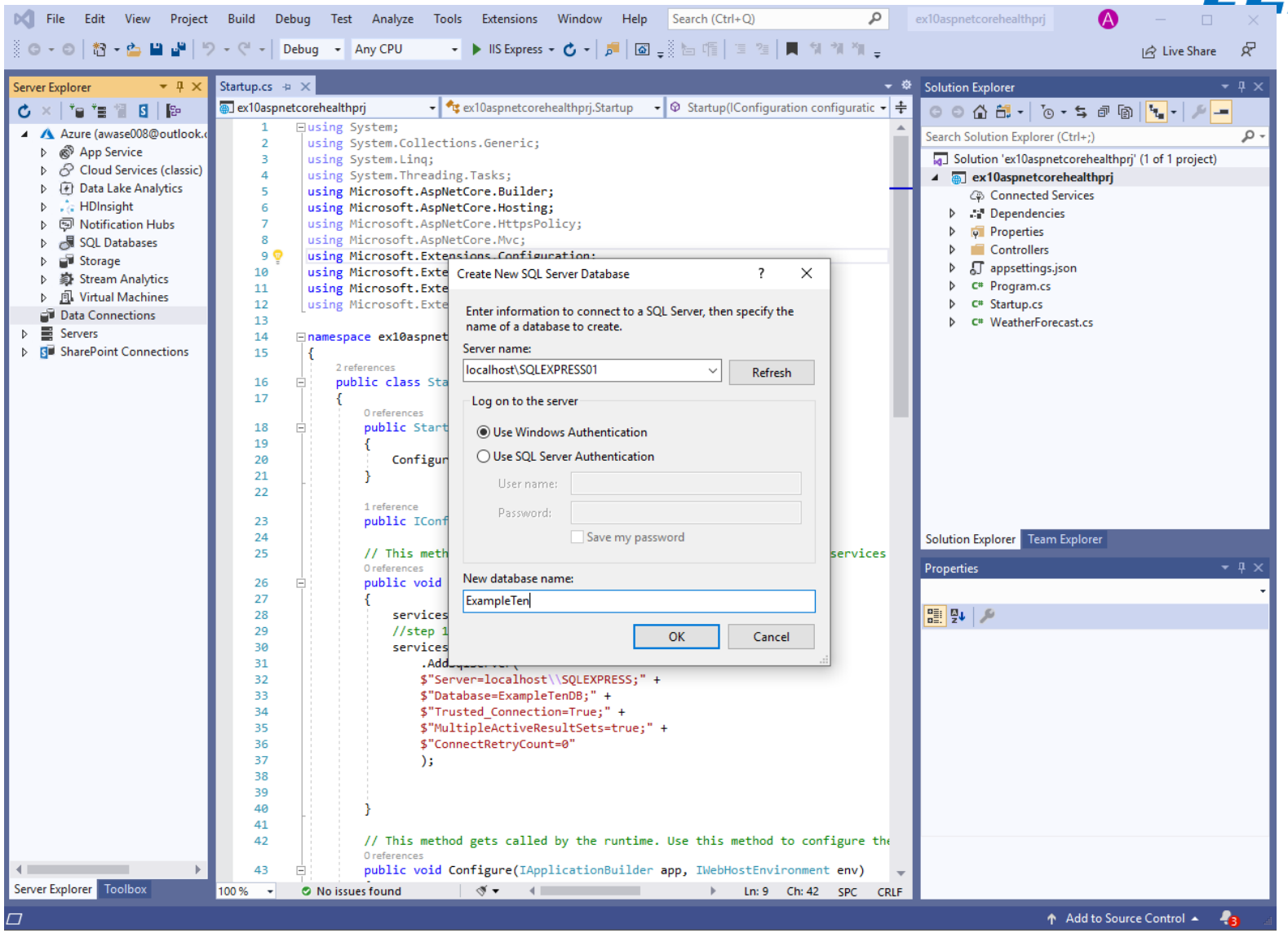
```
// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
0 references
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }

    app.UseHttpsRedirection();

    app.UseRouting();

    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllers();
        //step2: define endpoint to render health checks
        endpoints.MapHealthChecks("/apphealth");
    });
}
```





Example:

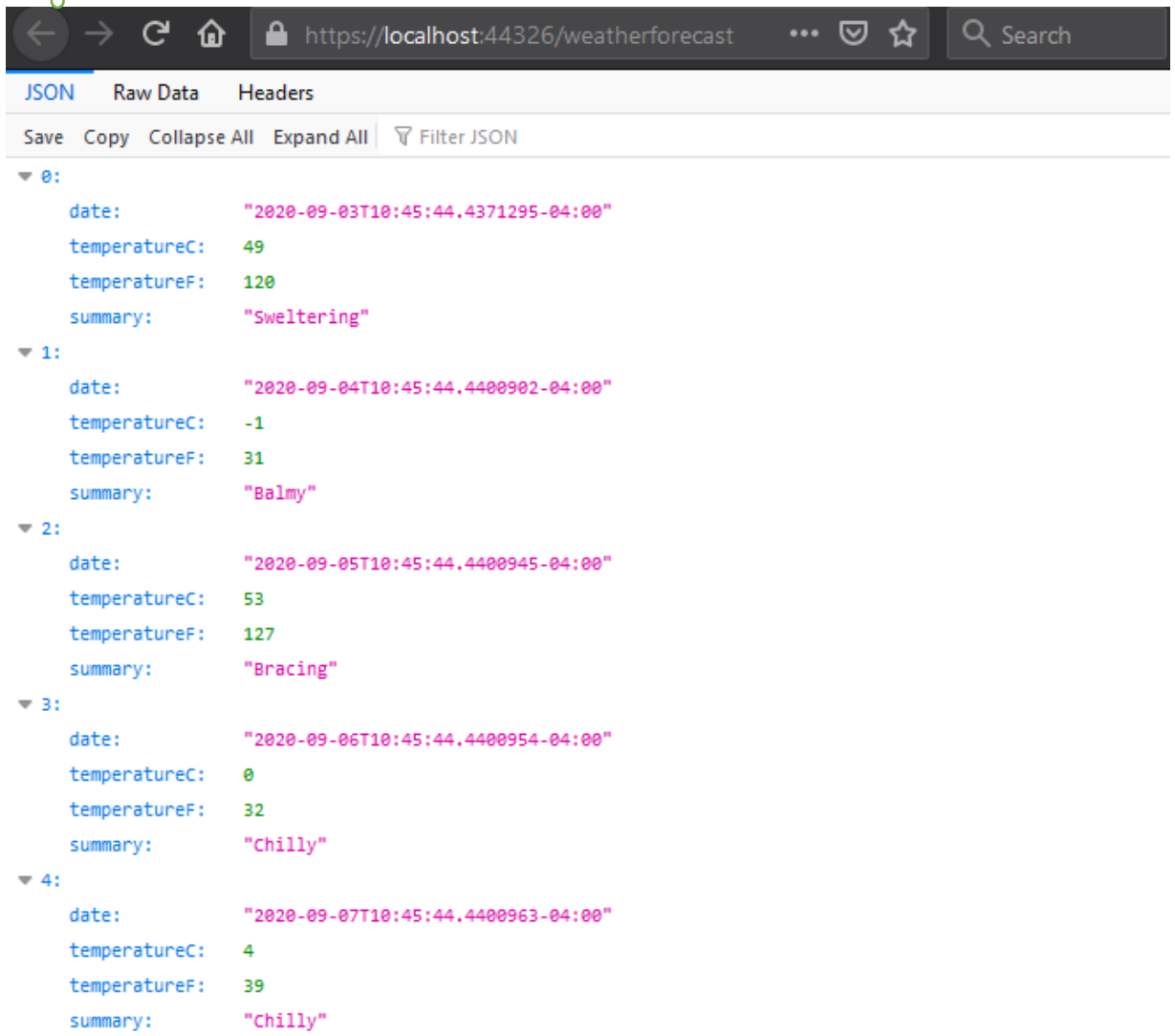




Original Series

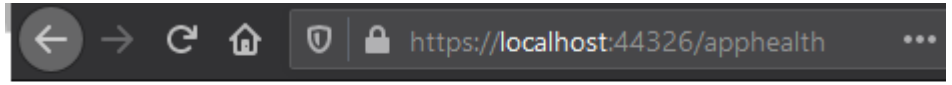
Example:

https://localhost:44326/weatherforecast



Example:

https://localhost:44326/apphealth




```
Unhealthy
```

Example:

dotnet tool install --global dotnet-counters --version 1.0.3-preview5.19251.2

```
PS C:\Users\Awase Khirni Syed> dotnet tool install --global dotnet-counters --version 1.0.3-preview5.19251.2

Welcome to .NET 5.0!
---------------------
SDK Version: 5.0.100-preview.7.20366.6

Telemetry
---------
The .NET tools collect usage data in order to help us improve your experience. The data is anonymous. It is collected by Microsoft and s
hared with the community. You can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variable to '1' or 'true'
using your favorite shell.

Read more about .NET CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

----------------
Installed an ASP.NET Core HTTPS development certificate.
To trust the certificate run 'dotnet dev-certs https --trust' (Windows and macOS only).
Learn about HTTPS: https://aka.ms/dotnet-https
----------------
Write your first app: https://aka.ms/dotnet-hello-world
Find out what's new: https://aka.ms/dotnet-whats-new
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cli
--------------------------------------------------------------------------------
You can invoke the tool using the following command: dotnet-counters
Tool 'dotnet-counters' (version '1.0.3-preview5.19251.2') was successfully installed.
```

Dotnet-counters --help

```
PS C:\Users\Awase Khirni Syed> dotnet-counters --help
Usage:
  dotnet-counters [options] [command]

Options:
  --version    Display version information

Commands:
  monitor <counter_list>    Start monitoring a .NET application
  list                      Display a list of counter names and descriptions, grouped by provider.
```

Example:

Dotnet tool install –global dotnet-trace

```
PS C:\Users\Awase Khirni Syed> dotnet tool install --global dotnet-trace
You can invoke the tool using the following command: dotnet-trace
Tool 'dotnet-trace' (version '3.1.141901') was successfully installed.
```

```
PS C:\Users\Awase Khirni Syed> dotnet-trace --help
Usage:
  dotnet-trace [options] [command]

Options:
  --version    Display version information

Commands:
  collect                   Collects a diagnostic trace from a currently running process
  ps                        Lists the dotnet processes that traces can be collected
  list-profiles             Lists pre-built tracing profiles with a description of what providers and filters are in each profile
  convert <input-filename>  Converts traces to alternate formats for use with alternate trace analysis tools. Can only convert from th
e nettrace format
```

```
PS C:\Users\Awase Khirni Syed> dotnet-trace ps
    126992 iisexpress C:\Program Files\IIS Express\iisexpress.exe
```

```
PS C:\Users\Awase Khirni Syed> dotnet-trace collect -p 126992
No profile or providers specified, defaulting to trace profile 'cpu-sampling'

Provider Name                            Keywords            Level               Enabled By
Microsoft-DotNETCore-SampleProfiler      0x0000F00000000000  Informational(4)    --profile
Microsoft-Windows-DotNETRuntime          0x00000014C14FCCBD  Informational(4)    --profile

Process        : C:\Program Files\IIS Express\iisexpress.exe
Output File    : C:\Users\Awase Khirni Syed\trace.nettrace
```

Example:


```
dotnet tool install --global dotnet-dump --version 1.0.3-preview5.19251.2
```

```
PS C:\Users\Awase Khirni Syed> dotnet tool install --global dotnet-dump --version 1.0.3-preview5.19251.2
You can invoke the tool using the following command: dotnet-dump
Tool 'dotnet-dump' (version '1.0.3-preview5.19251.2') was successfully installed.
```