**.NET Core 3.0, VS2019 and C# 8.0 for ASP .NET Core developers**

By Shahed C on April 8, 2019

[9 Replies](https://wakeupandcode.com/net-core-3-vs2019-and-csharp-8/#comments)

This is the **fourteenth**of a [series of posts](https://wakeupandcode.com/aspnetcore/#aspnetcore2019) on ASP .NET Core in 2019. In this series, we’ll cover 26 topics over a span of 26 weeks from January through June 2019, titled **A-Z of ASP .NET Core!**

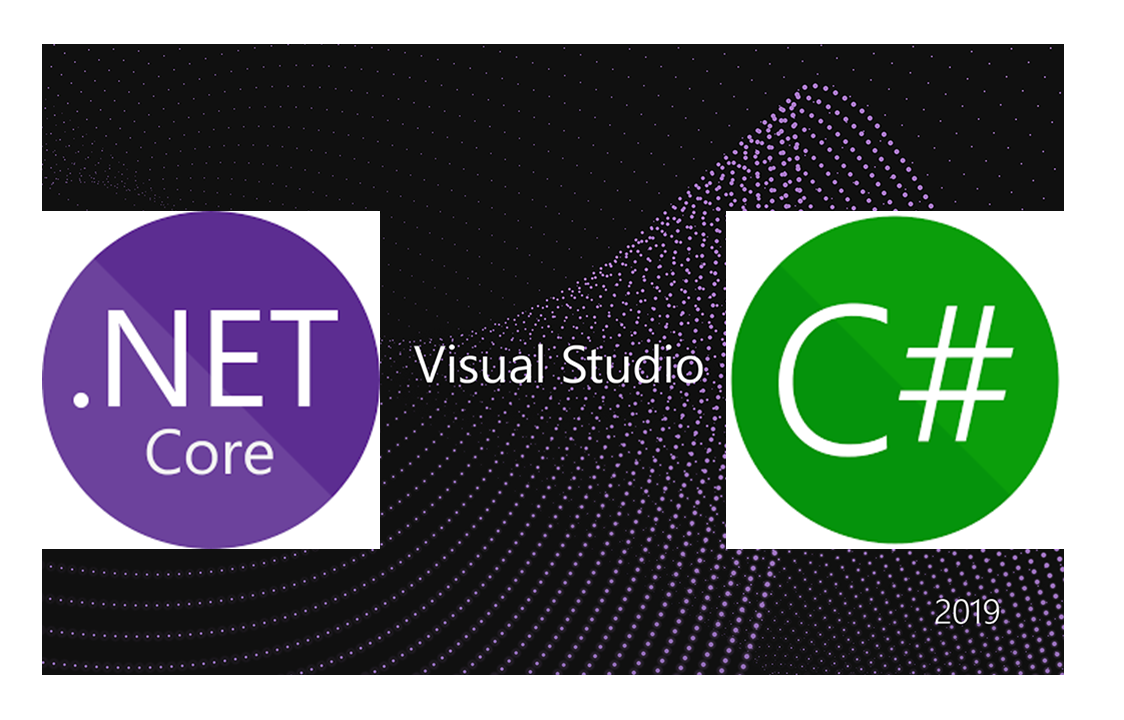
** A – Z of ASP .NET Core!**

**In this Article:**

* N is for .NET Core 3.0, VS2019 and C# 8.0
* Visual Studio 2019
* .NET Core 3.0
* C# 8.0 features
* ASP .NET Core 3.0
* References

**N is for .NET Core 3.0, VS2019 and C# 8.0**

After completing 13 blog posts in this A-Z series, Visual Studio 2019 has now been released. While this is just a coincidence, this is a great opportunity to focus on .NET Core 3.0, VS 2019 and C# 8.0 language features for ASP .NET Core developers. This blog post provides an overview of everything you need to know to get started with the above, as an ASP .NET Core developer.

[](https://wakeupandcode.com/wp-content/uploads/2019/04/vs2019-netcore3-csharp8.png)

**Visual Studio 2019**

First things first: now that Visual Studio 2019 has been released, where can you download it from? Start with the main download page and then select the edition you need:

* VS2019 Downloads: <https://visualstudio.microsoft.com/downloads/>
* [Community Edition](https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&amp;rel=16)
* [Professional Edition](https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Professional&amp;rel=16)
* [Enterprise Edition](https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Enterprise&amp;rel=16)

As before, the Community Edition (comparable to Pro) is free for students, open-source contributors and individuals. The Pro and Enterprise editions add additional products and services from small teams to enterprise companies.

*But wait!* What if you can’t stay online for the length of the installation or need to reinstall quickly at a later date? If you need an offline installer, check out the instructions on the following page:

* Create an offline installation: <https://docs.microsoft.com/en-us/visualstudio/install/create-an-offline-installation-of-visual-studio?view=vs-2019>

What are some cool new and improved features to be aware of? There are so many that I stitched together a series of tweets from [Amanda Silver](https://twitter.com/amandaksilver) (Director of Program Management for Dev Tools at Microsoft) and created the following thread:

* Twitter thread: <https://twitter.com/shahedC/status/1113177299652837376>

So many cool things in [#VS2019](https://twitter.com/hashtag/VS2019?src=hash&amp;ref_src=twsrc%5Etfw)! Here’s what’s awesome and new and improved!

Tweets from [@amandaksilver](https://twitter.com/amandaksilver?ref_src=twsrc%5Etfw) 👇

— Shahed (on Leave) (@shahedC) [April 2, 2019](https://twitter.com/shahedC/status/1113177299652837376?ref_src=twsrc%5Etfw)

The aforementioned thread highlights the following features. Click each hyperlink in the list below for more info on each.

* [Live Share](https://visualstudio.microsoft.com/services/live-share/): Available as an extension in VS Code, Live Share is installed by default with VS2019. Easily collaborate with other developers while coding in real-time!
* [Intellicode](https://docs.microsoft.com/en-us/visualstudio/intellicode/): Use AI to write better code. Choose to share what you want with others or keep things private.
* [Git-first workflows](https://devblogs.microsoft.com/visualstudio/get-to-code-how-we-designed-the-new-visual-studio-start-window/): Choose to create a new project from a source code repo or use a template. The new start window provides more options up front.
* [Debug search](https://devblogs.microsoft.com/visualstudio/enhanced-in-visual-studio-2019-search-for-objects-and-properties-in-the-watch-autos-and-locals-windows/): Search while debugging. Type in search filters in the Watch, Locals, and Autos panels.
* [Snapshot debugging](https://docs.microsoft.com/en-us/visualstudio/debugger/debug-live-azure-applications?view=vs-2019): Available in the Enterprise Edition, snapshot debugging allows you to get a snapshot of your app’s execution after deployment. This includes cloud deployments, Azure VMs and Kubernetes containers.
* [VS Search](https://twitter.com/amandaksilver/status/1113171409482596352): Dynamic search results include commands, menus, components and templates. Note that this was formerly know as *Quick Launch*.
* [App Service Debugging](https://twitter.com/amandaksilver/status/1113171680422236162): Attach the debugger to your app running in Azure App Service!
* [App Service Connectivity](https://twitter.com/amandaksilver/status/1113171890955202560): Connect your web app to Azure App Service with ease, including App Insights monitoring.
* [Azure Monitor](https://twitter.com/amandaksilver/status/1113172181163290624): Use Azure Monitor to get additional insight on your deployed app!

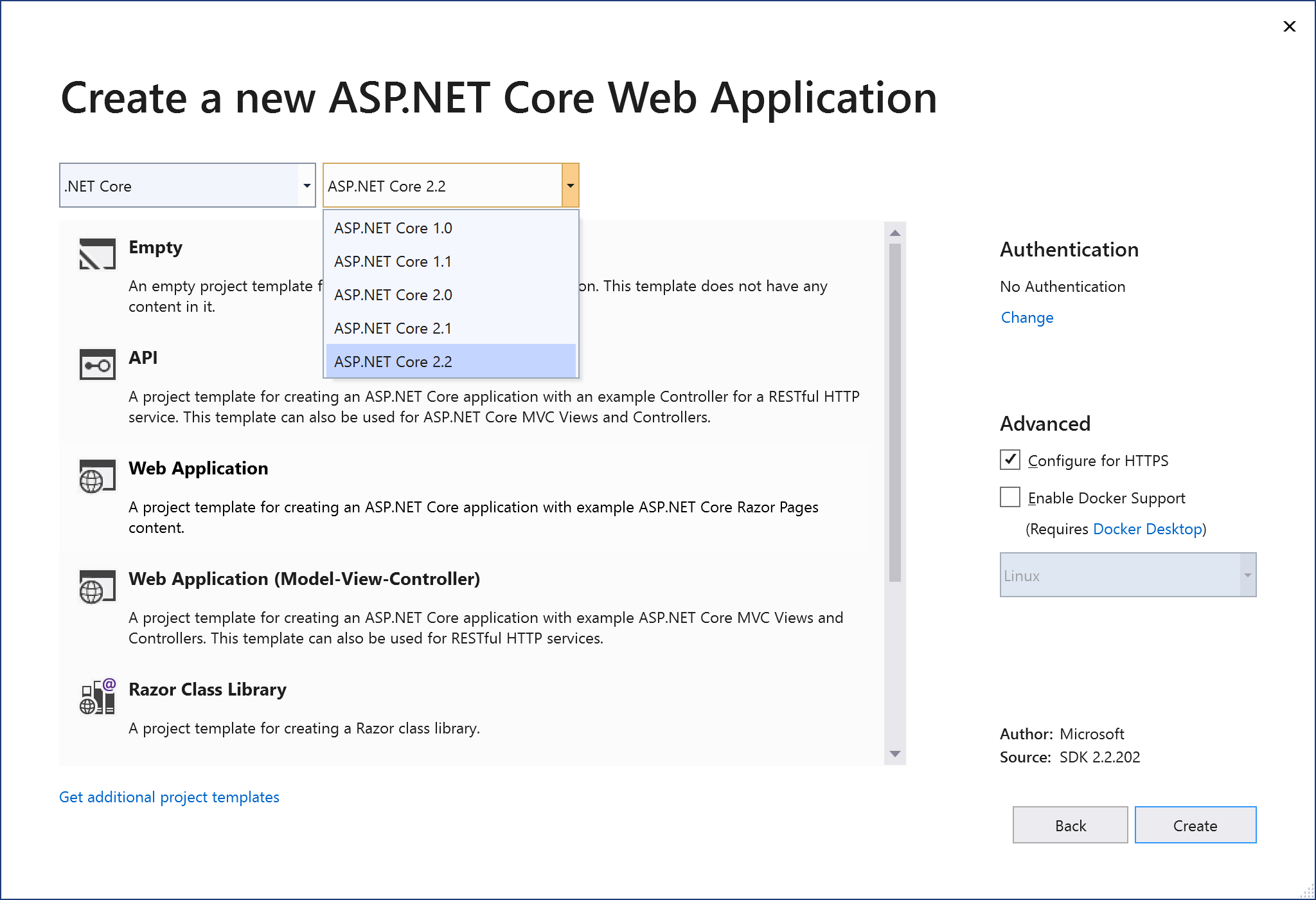
If you prefer to sit back and relax and just watch new product announcements, I put together a handy list of YouTube videos from the VS2019 launch event. This playlist kicks off with the 50-minute keynote, is followed by a string of videos and ends with a cumulative 7-hour video if you prefer to watch all at once.

* VS2019 Launch event playlist: [https://www.youtube.com/watch?v=DANLUUIUrcM&list=PLLPmfQOWse5Gr4pMcyFmwHW-hDUWmbT6B](https://www.youtube.com/watch?v=DANLUUIUrcM&amp;list=PLLPmfQOWse5Gr4pMcyFmwHW-hDUWmbT6B)

**.NET Core 3.0**

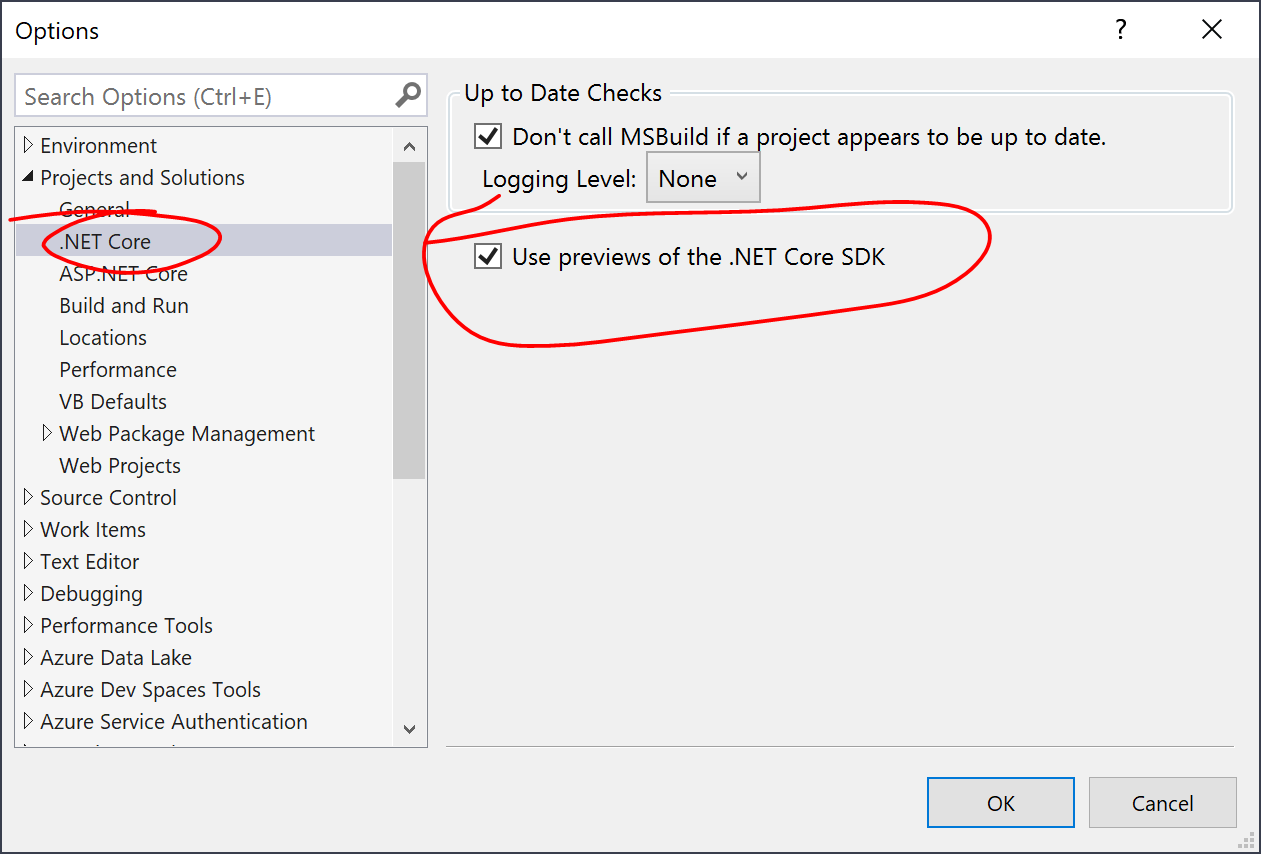
If you’ve downloaded Visual Studio 2019, you may have followed the following steps:

1. Download Visual Studio 2019
2. Click **File** | **New** | **Project**(or create new from splash screen)
3. Create a new ASP .NET Core Web App
4. Select .NET Core 3.0 as the project type/platform… *right?*

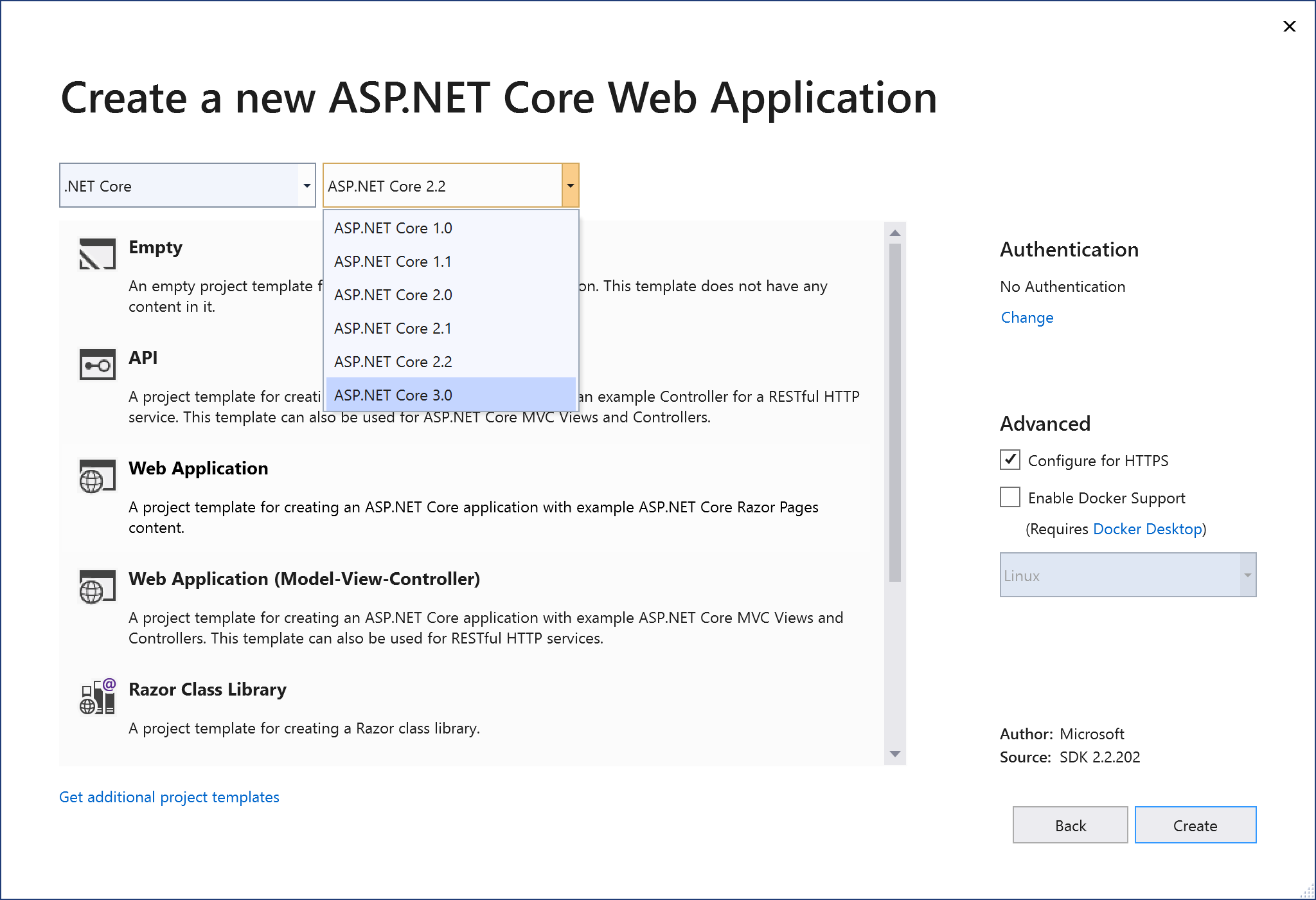
[](https://wakeupandcode.com/wp-content/uploads/2019/04/vs2019-new-aspnetcore.png)

*But wait a minute*, where is the option for ASP .NET Core 3.0? Why isn’t it available for selection? The answer is simple: ASP .NET Core 3.0 is still in preview as of April 2019, after the release of Visual Studio 2019. In order to create ASP .NET Core 3.0 projects with VS2019, you should do the following (for now):

* Download .NET Core 3.0: <https://dotnet.microsoft.com/download/dotnet-core/3.0>
* Enable preview releases of .NET Core
* Click **Tools** | **Options** in the top menu
* Expand **Projects and Solutions** | **.NET Core**
* Ensure that “Use previews of the .NET Core SDK” checkbox is checked

[](https://wakeupandcode.com/wp-content/uploads/2019/04/vs2019-netcore3-preview.png)

Start to create a new project again and you should now see an option for Core 3.0!

[](https://wakeupandcode.com/wp-content/uploads/2019/04/vs2019-new-aspnetcore3.png)

Following some Twitter feedback, here are my results from running MSBuild from a command line:

* Tweet: <https://twitter.com/shahedc/status/1115360210779017216>

Just tried this:  
> MSBuild.exe c:pathprojfile.csproj -t:rebuild

Results:  
Build succeeded.  
 0 Warning(s)  
 0 Error(s)

FYI, I used MSBuild.exe from C:Program Files (x86)Microsoft Visual Studio2019CommunityMSBuildCurrentBin

— Shahed (on Leave) (@shahedC) [April 8, 2019](https://twitter.com/shahedC/status/1115360210779017216?ref_src=twsrc%5Etfw)

Here is the specific command, setting the target to rebuild the project:

> MSBuild.exe c:\path\projfile.csproj -t:rebuild

On my development machine, I used MSBuild.exe from the following path:

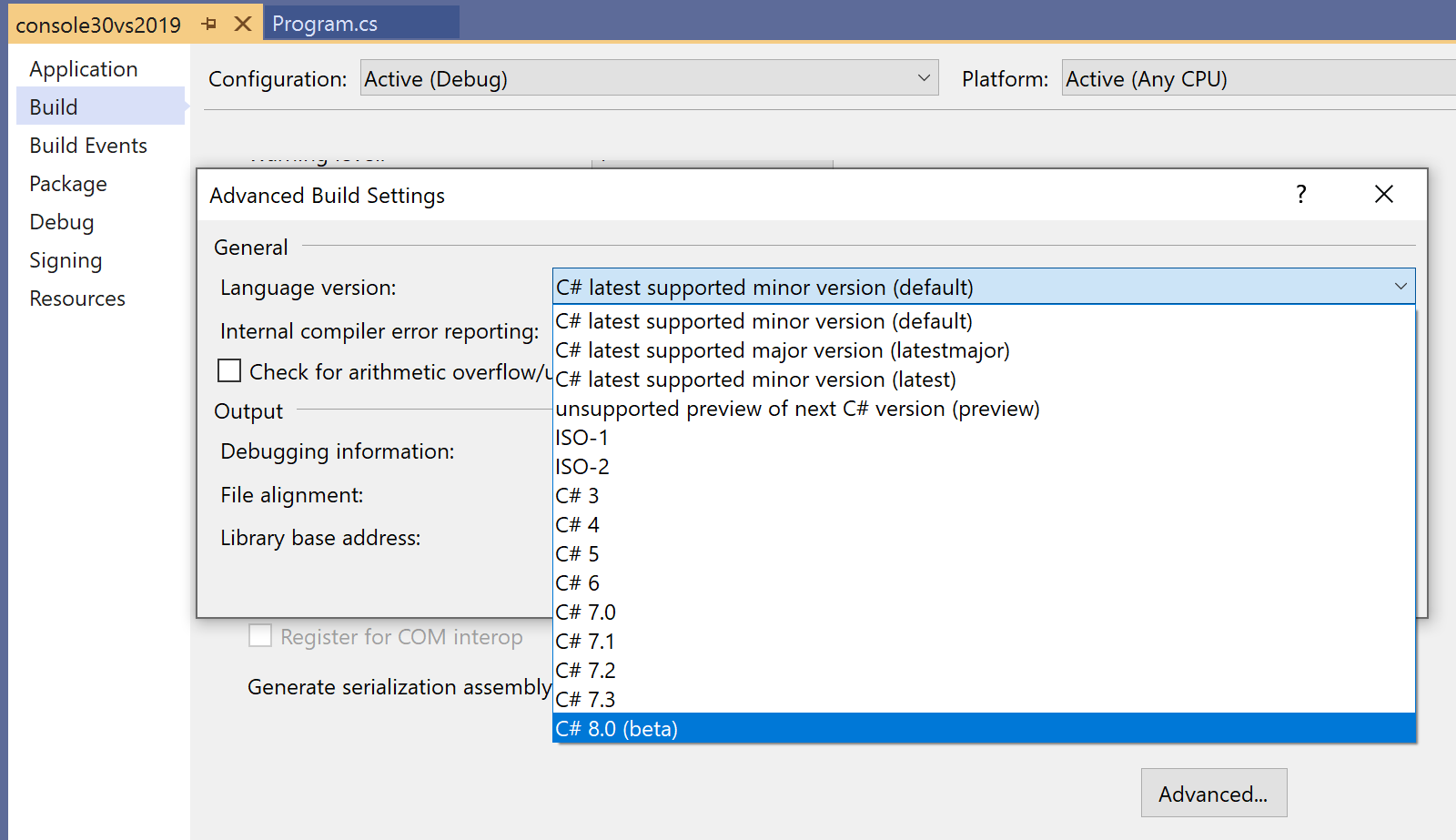
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\MSBuild\Current\Bin

**C# 8.0 Features**

To ensure that C# 8.0 preview features are available, you can set the **LangVersion** property explicitly for your project. This setting is buried deep inside your Advanced settings within your project’s Build tab.

To update the language setting:

1. Right-click your project in **Solution Explorer**.
2. Select **Properties** to view your project properties.
3. Click the **Build** tab within your project properties.
4. Click the **Advanced** button on the lower right.
5. Select the appropriate **Language version**, e.g. C# 8.0 (beta)
6. ***Optional***: you may select “**unsupported preview**…” instead

[](https://wakeupandcode.com/wp-content/uploads/2019/04/vs2019-build-advanced-lang.png)

The above screenshots show the aforementioned setting in the Visual Studio UI. If you wish to update your .csproj file directly, you may view/edit the <**LangVersion**> value. A few samples are shown below:

For a .NET Core 3.0 console app set to use C# *preview* versions, the value of <**LangVersion**> is set to the value “preview”:

<Project Sdk="Microsoft.NET.Sdk">  
 <PropertyGroup>  
 <OutputType>Exe</OutputType>  
 <TargetFramework>netcoreapp3.0</TargetFramework>  
 <LangVersion>preview</LangVersion>  
 </PropertyGroup>  
</Project>

For a .NET Core 3.0 console app set to use C# 8.0 explicitly, the value of <**LangVersion**> is set to the value “8.0”:

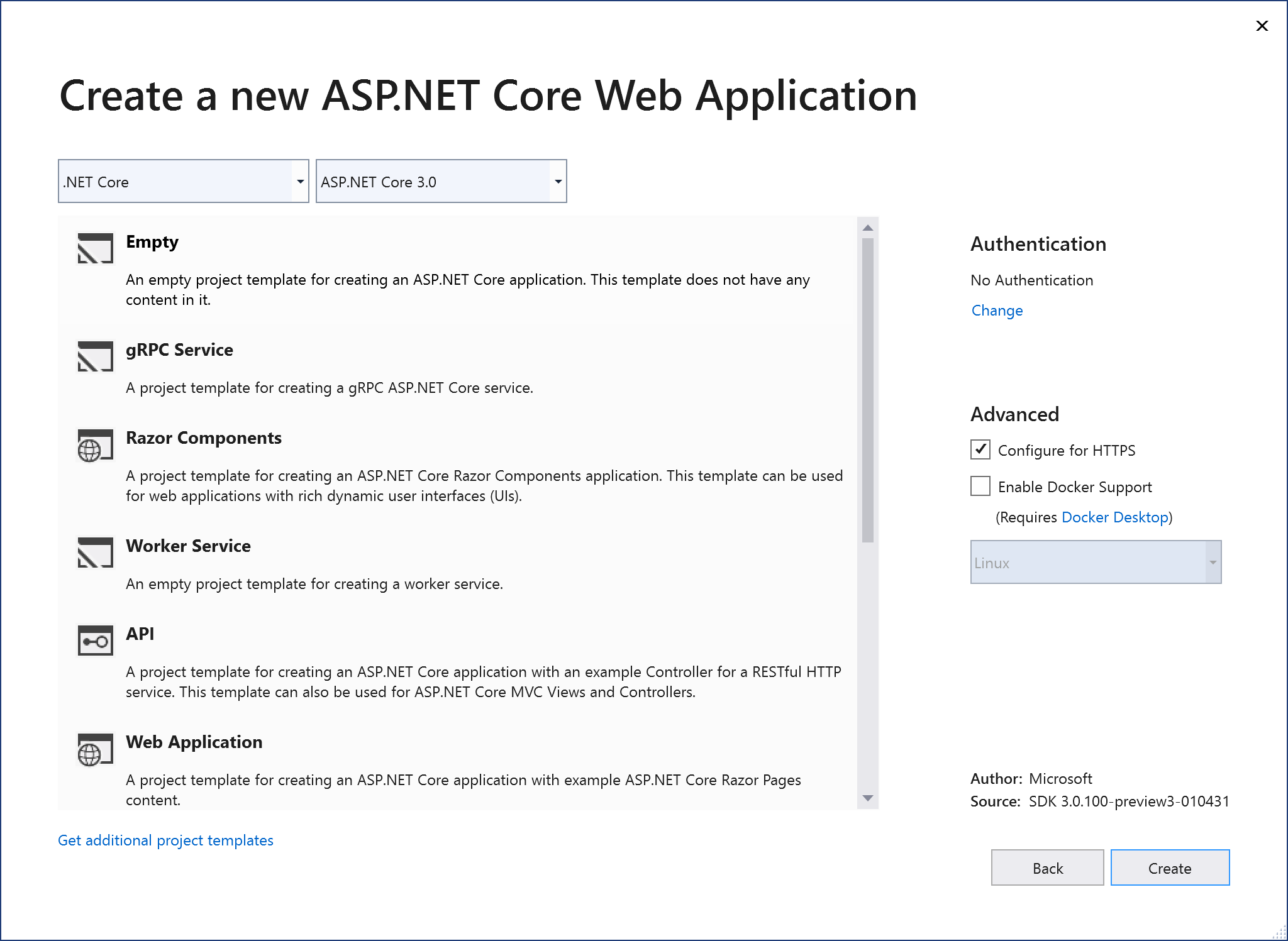
<Project Sdk="Microsoft.NET.Sdk">  
 <PropertyGroup>  
 <OutputType>Exe</OutputType>  
 <TargetFramework>netcoreapp3.0</TargetFramework>  
 <LangVersion>8.0</LangVersion>  
 </PropertyGroup>  
</Project>

Here is a list of C# 8.0 features, from recent preview releases:

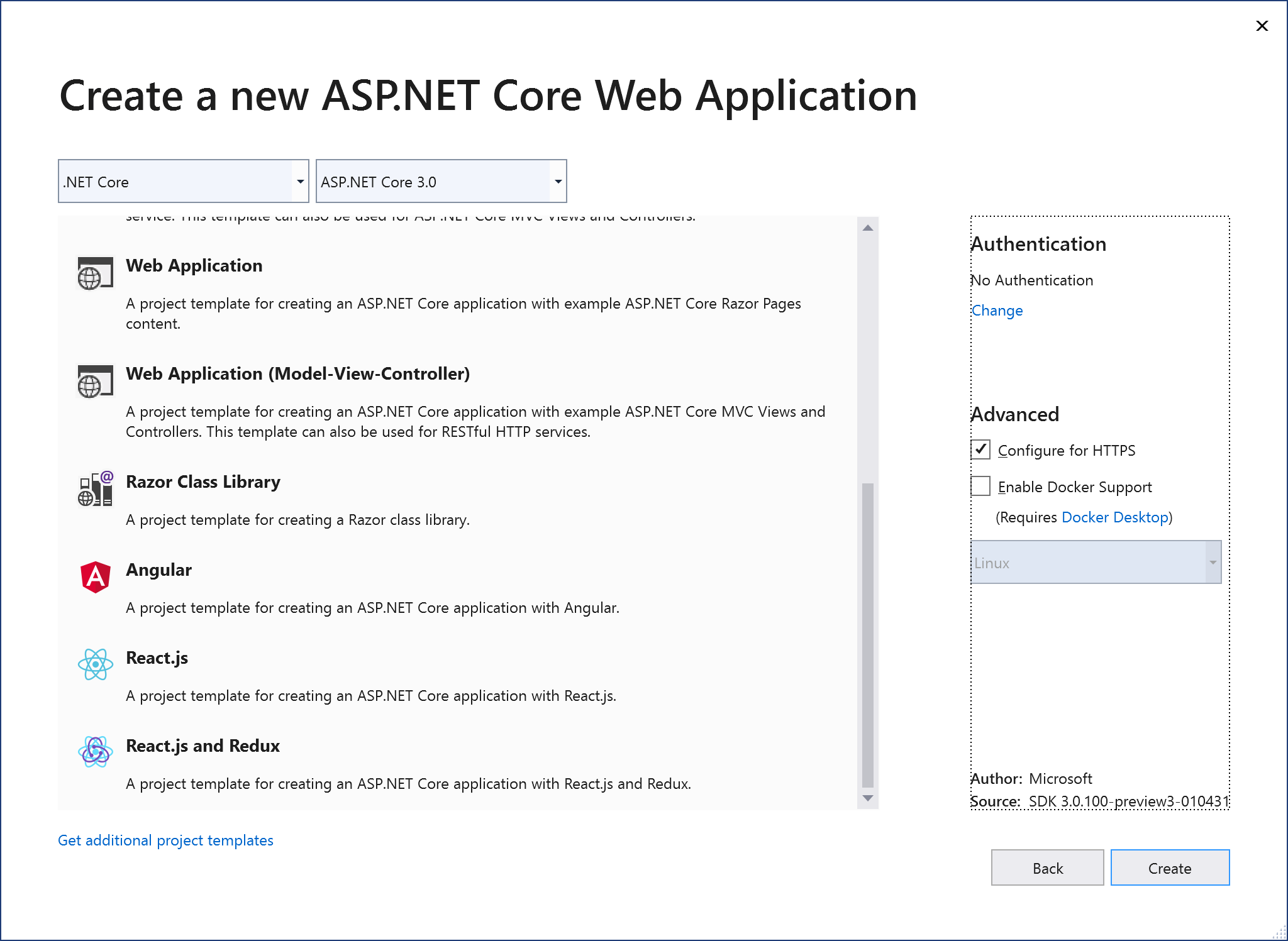
* [Nullable reference types](https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/nullable-reference-types): design with intent, decide that some variables must always have a value, while others may be missing a value, using nullable reference types.
* [Asynchronous streams](https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/generate-consume-asynchronous-stream): create and consume async streams, e.g. large streams of data
* [Indices and ranges](https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-8#indices-and-ranges): Specify subranges of data with Span<T>, indicate indices in the subsets of data
* [Pattern matching enhancements](https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-8#more-patterns-in-more-places):
* Switch expressions: replace repetitive switch-case blocks with simpler *switch expressions*
* Property patterns: enhance switch statements by matching an object’s properties
* Tuple patterns: use tuples for cases when matching sets of values
* Positional patterns: used to simplify the way we apply recursive patterns without having to name an object’s properties
* [Using declarations](https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-8#using-declarations): used to simplify using blocks (within which an object is disposed when done) by disposing of the object at the end of the *enclosing* scope, i.e. its parent block.
* [Static local functions](https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-8#static-local-functions): useful for local methods that are intended to be static.
* [Disposable ref structs](https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-8#disposable-ref-structs): allows the use of Dispose() methods to allow implementation of IDisposable in structs declared with a ref modifier.

**ASP .NET Core 3.0 Project Types**

When you create a new Core 3.0 web project with Visual Studio 2019, you’ll see some familiar project types. You will also see some *new* project types. These are shown in the 2 screenshots below:

[](https://wakeupandcode.com/wp-content/uploads/2019/04/vs2019-web-projects01.png)

Web Projects 1 of 2

[](https://wakeupandcode.com/wp-content/uploads/2019/04/vs2019-web-projects02.png)

Web Projects 2 of 2

The above project types are described below:

1. **Empty**: familiar empty project that just writes out Hello World to the HTTP Response, *without* the use of MVC or Razor Pages
2. **gRPC Service**: a *new* project type using Google’s high-performance Remote Procedure Call (RPC) framework
3. **Razor Components**: initially called server-side Blazor, renamed to Razor Components to distinguish it from client-side Blazor, this will be once again be [renamed to server-side Blazor again when ASP .NET Core 3.0 is released](https://github.com/aspnet/AspNetCore/issues/8931). Allows full-stack C# web app development.
4. **Worker Service**: a *new*project type that allows creation of background processes, e.g. Windows services or Linux daemons. May be relocated in the template list upon release.
5. **API**: familiar project type for creating Web APIs and RESTful services. Can be mixed and matched with Razor Pages or MVC components.
6. **Web Application**: familiar project type for creating Web Apps with Razor Pages. Can be mixed and matched with Web API and/or MVC components.
7. **Web Application (MVC)**: familiar project type for creating Web Apps with MVC application structure. Can be mixed and matched with Razor Pages and/or Web API.
8. **Razor Class Library**: relatively new project type for creating reusable UI Class Libraries with Razor Pages. See previous post on [Razor Class Libraries](https://wakeupandcode.com/your-first-razor-ui-library-with-asp-net-core/).
9. **Angular, React.js, React.js and Redux**: familiar web projects for web developers who wish to build a JavaScript front-end, typically with a Web API backend.

Well, what about *client-side Blazor*? You may have noticed that server-side Blazor (aka Razor Components are mentioned, but there is no sign of client-side Blazor. As of April 2019, client-side Blazor running in the browser with WebAssembly is still experimental. As a result, it is not included with ASP .NET Core 3.0 but can be downloaded separately.

* Blazor Extension: <https://marketplace.visualstudio.com/items?itemName=aspnet.blazor>

For a quick refresher, check out my previous post on client-side Blazor:

* Blazor Full-Stack Web Dev in ASP .NET Core: <https://wakeupandcode.com/blazor-full-stack-web-dev-in-asp-net-core/>

If you need some help getting started, here’s a handy guide from the official docs:

* Tutorial: Create an ASP.NET Core web app with Entity Framework and Visual Studio 2019: <https://docs.microsoft.com/en-us/visualstudio/get-started/csharp/tutorial-aspnet-core-ef-step-01?view=vs-2019>

**References**

* What’s new in .NET Core 3.0: <https://docs.microsoft.com/en-us/dotnet/core/whats-new/dotnet-core-3-0>
* A first look at changes coming in ASP.NET Core 3.0: <https://devblogs.microsoft.com/aspnet/a-first-look-at-changes-coming-in-asp-net-core-3-0/>
* ASP.NET Core updates in .NET Core 3.0 Preview 2: <https://devblogs.microsoft.com/aspnet/aspnet-core-3-preview-2/>
* ASP.NET Core updates in .NET Core 3.0 Preview 3: <https://devblogs.microsoft.com/aspnet/asp-net-core-updates-in-net-core-3-0-preview-3/>
* Blazor 0.9.0 experimental release now available: <https://devblogs.microsoft.com/aspnet/blazor-0-9-0-experimental-release-now-available/>
* Visual Studio 2019 Launch Event: <https://visualstudio.microsoft.com/vs2019-launch/>
* What’s new in Visual Studio 2019: https://docs.microsoft.com/en-us/visualstudio/ide/whats-new-visual-studio-2019?view=vs-2019
* What’s New in C# 8.0: <https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-8>
* Little great things about Visual Studio 2019: <https://devblogs.microsoft.com/visualstudio/little-great-things-about-visual-studio-2019/>
* Do more with patterns in C# 8.0: <https://devblogs.microsoft.com/dotnet/do-more-with-patterns-in-c-8-0/>
* Get started with gRPC in ASP.NET Core: <https://docs.microsoft.com/en-us/aspnet/core/tutorials/grpc/grpc-start>