**Hello ASP .NET Core v3.1!**

By Shahed C on December 23, 2019

[4 Replies](https://wakeupandcode.com/hello-asp-net-core-v3-1/#comments)

Last week, I [started blogging again](https://wakeupandcode.com/asp-net-core-code-sharing-between-blazor-mvc-and-razor-pages/) after a few months off, thanks to the [3rd Annual C# Advent Series](https://crosscuttingconcerns.com/The-Third-Annual-csharp-Advent). Before I get into my new and improved A-Z series for 2020, here’s a quick refresh on how to get started with ASP .NET Core 3.1! This is an updated version of [2018’s “Hello ASP .NET Core!” post](https://wakeupandcode.com/hello-asp-net-core/).

In This Article:

* Why ASP .NET Core?
* Dev Tools for ASP .NET Core
* Project Types
* Tutorials and Resources
* Blogs and Video Streams to Follow

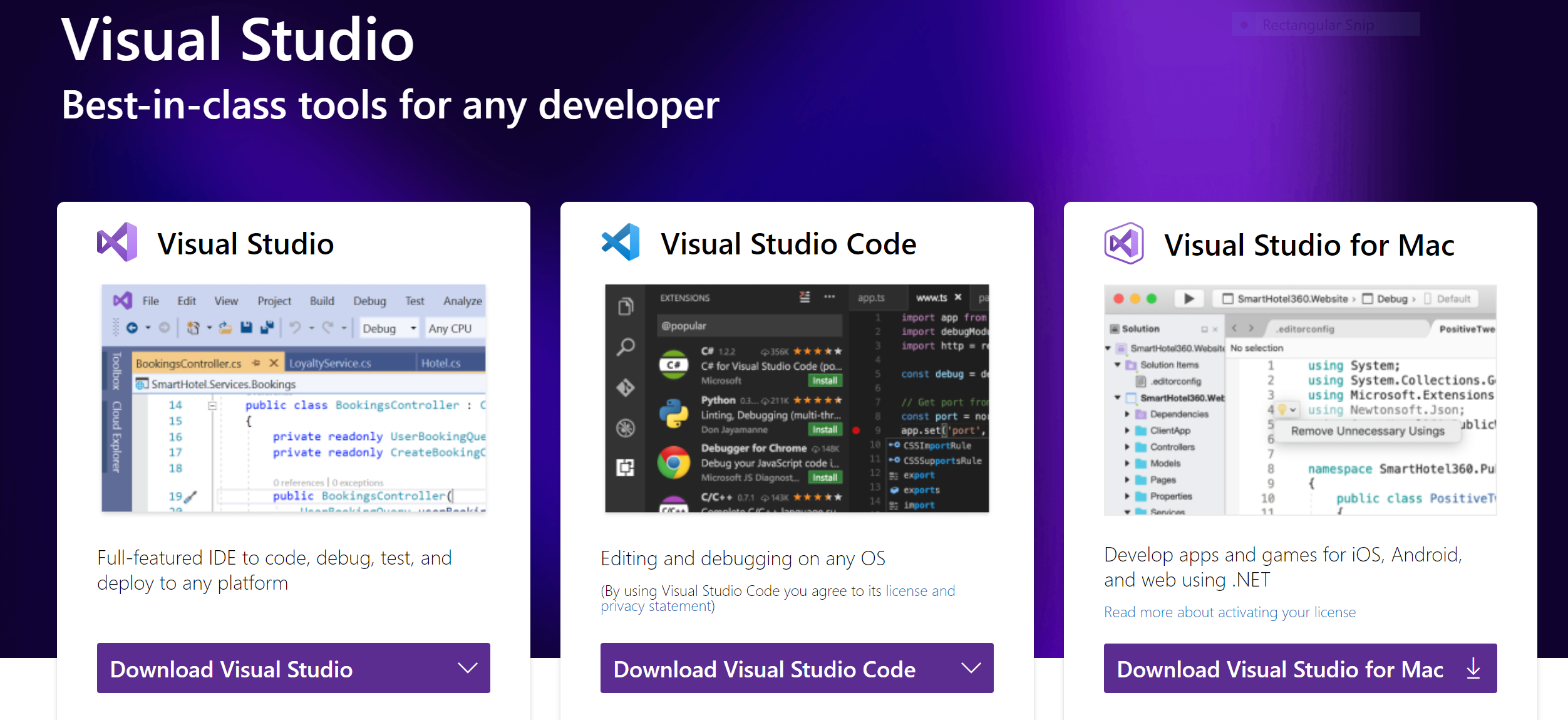
**Why ASP .NET Core?**

Throughout my [2018 mini-series](https://wakeupandcode.com/aspnetcore/#aspnetcore2018) and [2019 A-Z series](https://wakeupandcode.com/aspnetcore/#aspnetcore2019), I’ve regularly kept up to date on what’s new with ASP .NET Core for building Web Apps, Web APIs and even full-stack C# web applications with [Blazor](https://docs.microsoft.com/en-us/aspnet/core/blazor)!

With the 2019 release of ASP .NET Core 3.0 and the LTS follow-up ASP .NET Core 3.1, now is still a great time to be an ASP .NET Core developer. But where should you begin? You have many options.

**Dev Tools for ASP .NET Core**

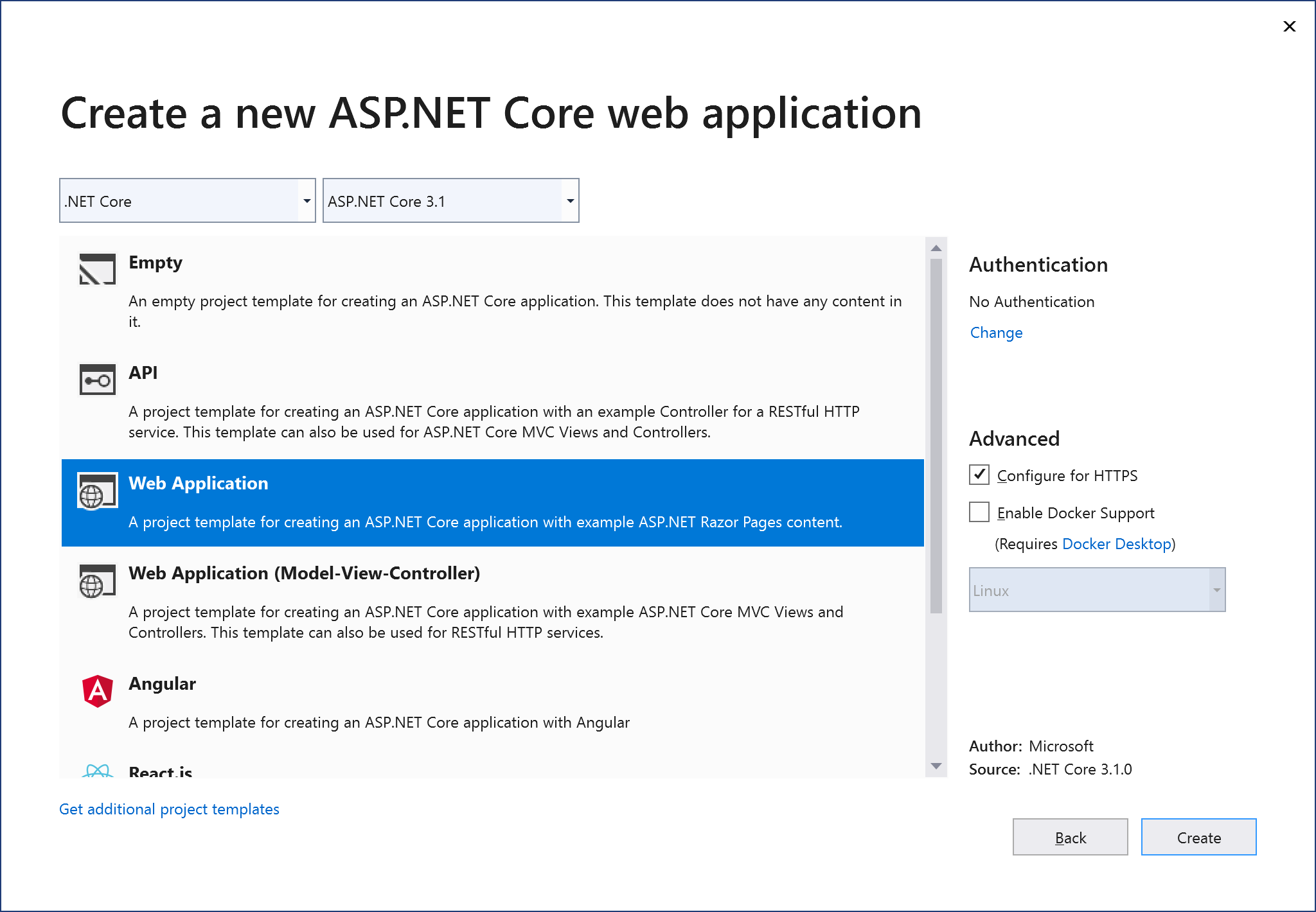
1. [Visual Studio for Windows](https://visualstudio.microsoft.com/vs/): a full-fledged IDE that runs on Windows (or even on Windows on your Mac via Parallels/Bootcamp. Get the latest version to try out new stable ASP .NET features and get the Preview versions to try cool new features. Install just the pieces you need. Start with the free Community edition for students, individuals and open-source projects.
2. [Visual Studio for Mac](https://visualstudio.microsoft.com/vs/mac/): build ASP .NET Core applications on a Mac in a full-fledged IDE. VS for Mac can also be used for macOS apps, Xamarin mobile apps for iOS/Android, cloud apps and more!
3. [Visual Studio Code](https://code.visualstudio.com): super light-weight code editor available on Windows, MacOS and Linux. Use the dotnet CLI tools in the integrated Terminal to quickly build ASP .NET Core web apps and more!

[](https://visualstudio.microsoft.com/)Visual Studio downloads

**Project Types**

Once you create a new ASP .NET Core 3.1 Web project, you’ll have several options, including some new/recent ones:

1. **Empty**: start with a clean slate if you know what you’re doing.
2. **API**: start with a sample Web API endpoint, with the ability to add MVC components, Razor web pages or Blazor as needed.
3. **Web Application**: start with a sample Web App that contains Razor web pages, with the ability to add Web APIs and MVC components as needed.
4. **Web Application (MVC)**: start with a sample Web App that contains MVC components (Models, Views and Controllers), with the ability to add Web APIs and Razor web pages as needed.
5. **Angular**: build an ASP .NET Core application with an Angular front-end.
6. **React.js:** build an ASP .NET Core application with a ReactJS front-end.
7. **React.js and Redux:** build an ASP .NET Core application with a ReactJS + Redux front-end.

[](https://wakeupandcode.com/wp-content/uploads/2019/12/vs2019-newproject-aspnetcore-web.png)ASP .NET Core New Project Templates

Other useful project types include (but not limited to):

1. **Blazor App**: currently includes only a Blazor Server App template as of v3.1, with client-side Blazor coming soon (2020). You can still use a preview-version of [client-side Blazor](https://docs.microsoft.com/en-us/aspnet/core/blazor/hosting-models?view=aspnetcore-3.1#blazor-webassembly) if you want to experiment with it now.
2. **Razor Class Library**: introduced in v2.1, build a reusable UI class library.
3. **Class Library**: Target .NET Core 3.1 or .NET Standard (skip .NET Framework libraries for your Core projects)

Note that HTTPS is selected by default and is [extremely easy](https://www.hanselman.com/blog/DevelopingLocallyWithASPNETCoreUnderHTTPSSSLAndSelfSignedCerts.aspx) to include in an ASP .NET Core web application, even in a development environment. Optionally, select a [Docker environment](https://docs.docker.com/engine/examples/dotnetcore/) and Authentication type as appropriate.

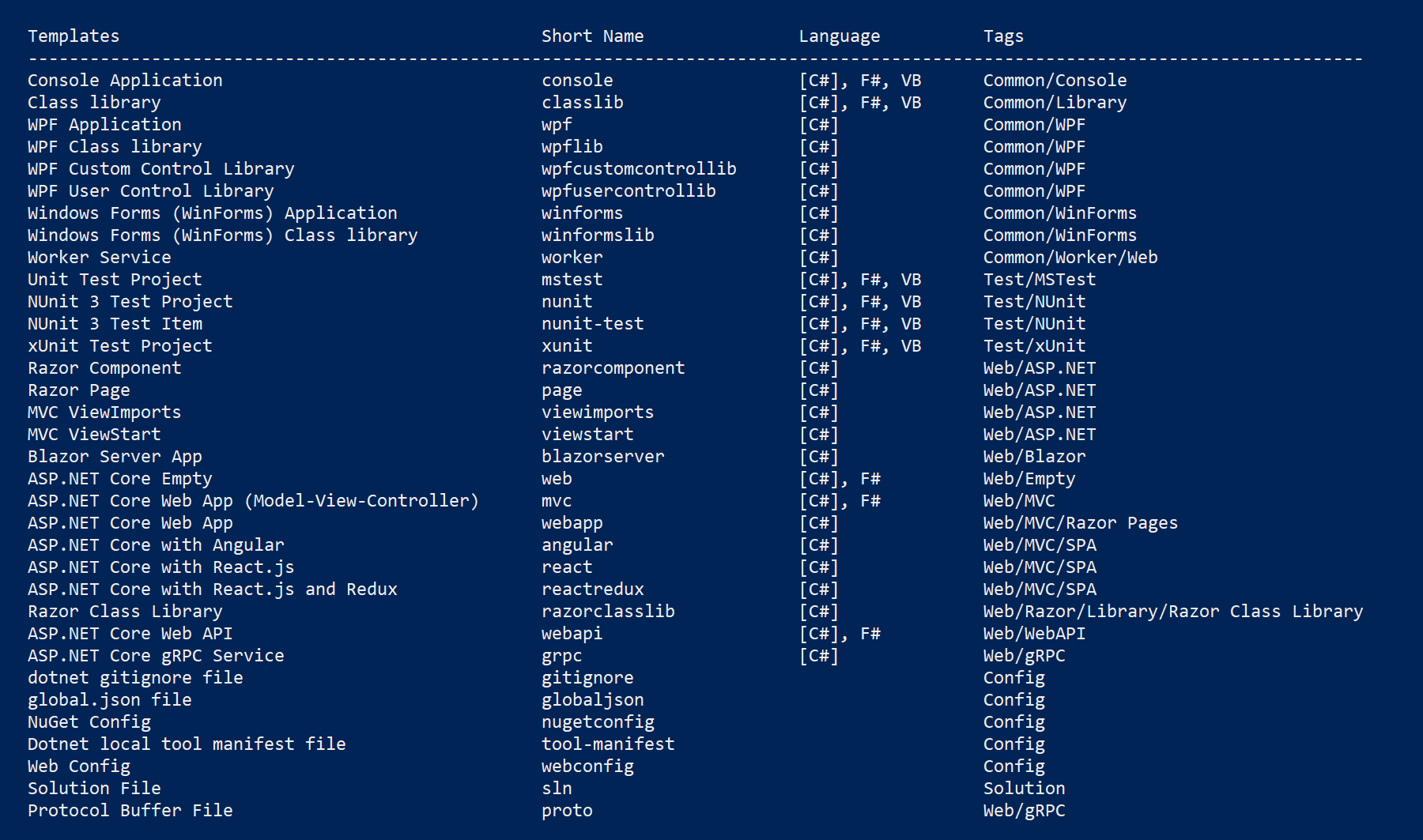
When using Visual Studio Code or the [dotnet CLI tools](https://docs.microsoft.com/en-us/dotnet/core/tools/dotnet-new?tabs=netcore21), you can create the same project types using CLI commands:

dotnet new <TEMPLATE> [--force] [-i|--install] [-lang|--language] [-n|--name] [--nuget-source] [-o|--output] [-u|--uninstall] [Template options]   
  
dotnet new <TEMPLATE> [-l|--list] [--type]   
  
dotnet new [-h|--help]

Some additional parameters are dry-run, update-check and update-apply. The parameters and their documented descriptions are listed below:

--dry-run: "Displays a summary of what would happen if the given command line were run if it would result in a template creation."  
  
--update-check: "Check the currently installed template packs for updates."  
  
--update-apply: "Check the currently installed template packs for update, and install the updates."

The options for new project templates are shown below:

[](https://wakeupandcode.com/wp-content/uploads/2019/12/dotnet-new-templates.png)dotnet new project templates

**Tutorials and Resources**

The 2020 A-Z blog series will go into specific topics within ASP .NET Core. In the meantime, check out this list of tutorials and resources to help you get started.

* MSFT docs: <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-3.1>
* Blazor: <https://docs.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-3.1>
* YouTube: <https://aka.ms/aspnetcore101-2019>
* Pluralsight: <https://www.pluralsight.com/search?q=asp.net%20core>
* Build Conf (May) 2019: <https://aka.ms/aspnetcore101-2019>
* .NET Conf  (Sep) 2019: <https://aka.ms/AspNetCore-DotNetConf2019>

**Blogs and Video Streams to Follow**

Here is a list of blogs and videos streams to follow for .NET Core and ASP .NET Core:

1. .NET Blog: <https://blogs.msdn.microsoft.com/dotnet>
2. ASP .NET Blog: <https://devblogs.microsoft.com/aspnet/>
3. Scott Hanselman: <https://www.hanselman.com/blog>
4. Iris Classon: <http://irisclasson.com>
5. Andrew Lock: <https://andrewlock.net>
6. Mike’s dotnetting: <https://www.mikesdotnetting.com>
7. Talking Dotnet: <http://www.talkingdotnet.com>
8. Dotnetcore tutorials: <https://dotnetcoretutorials.com>
9. ASP .NET Hacker: <http://asp.net-hacker.rocks>
10. Morning Brew: <http://blog.cwa.me.uk>
11. David Pine: <https://davidpine.net>
12. Steve Gordon: <https://www.stevejgordon.co.uk>
13. ASP .NET Community Standup: <https://live.asp.net>
14. Jeff Fritz on Twitch: <https://www.twitch.tv/csharpfritz>
15. .NET Core Podcast: <https://dotnetcore.show/>
16. Tim Corey on YouTube: <https://www.youtube.com/IAmTimCorey/>

Enjoy!