1. Intro:

* What is ASP.NET Core?

ASP.NET Core is a cross-platform, high-performance, [open-source](https://github.com/aspnet/home) framework for building modern, cloud-based, Internet-connected applications. With ASP.NET Core, you can:

* Build web apps and services, [IoT](https://www.microsoft.com/internet-of-things/) apps, and mobile backends.
* Use your favorite development tools on Windows, macOS, and Linux.
* Deploy to the cloud or on-premises.
* Run on [.NET Core or .NET Framework](https://docs.microsoft.com/dotnet/articles/standard/choosing-core-framework-server).
* Why use ASP.NET Core?

Millions of developers have used (and continue to use) [ASP.NET 4.x](https://docs.microsoft.com/en-us/aspnet/overview) to create web apps. ASP.NET Core is a redesign of ASP.NET 4.x, with architectural changes that result in a leaner, more modular framework.

ASP.NET Core provides the following benefits:

* A unified story for building web UI and web APIs.
* Integration of [modern, client-side frameworks](https://docs.microsoft.com/en-us/aspnet/core/client-side/index) and development workflows.
* A cloud-ready, environment-based [configuration system](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/configuration/index).
* Built-in [dependency injection](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection).
* A lightweight, [high-performance](https://github.com/aspnet/benchmarks), and modular HTTP request pipeline.
* Ability to host on [IIS](https://docs.microsoft.com/en-us/aspnet/core/publishing/iis), [Nginx](https://docs.microsoft.com/en-us/aspnet/core/publishing/linuxproduction), [Apache](https://docs.microsoft.com/en-us/aspnet/core/publishing/apache-proxy), [Docker](https://docs.microsoft.com/en-us/aspnet/core/publishing/docker), or self-host in your own process.
* Side-by-side app versioning when targeting [.NET Core](https://docs.microsoft.com/dotnet/articles/standard/choosing-core-framework-server).
* Tooling that simplifies modern web development.
* Ability to build and run on Windows, macOS, and Linux.
* Open-source and [community-focused](https://live.asp.net/).

ASP.NET Core ships entirely as [NuGet](https://www.nuget.org/) packages. This allows you to optimize your app to include only the necessary NuGet packages. In fact, ASP.NET Core 2.x apps targeting .NET Core only require a [single NuGet package](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/metapackage). The benefits of a smaller app surface area include tighter security, reduced servicing, and improved performance.

* Build web APIs and web UI using ASP.NET Core MVC

ASP.NET Core MVC provides features to build [web APIs](https://docs.microsoft.com/en-us/aspnet/core/tutorials/index#building-web-apis) and [web apps](https://docs.microsoft.com/en-us/aspnet/core/tutorials/index#building-web-applications):

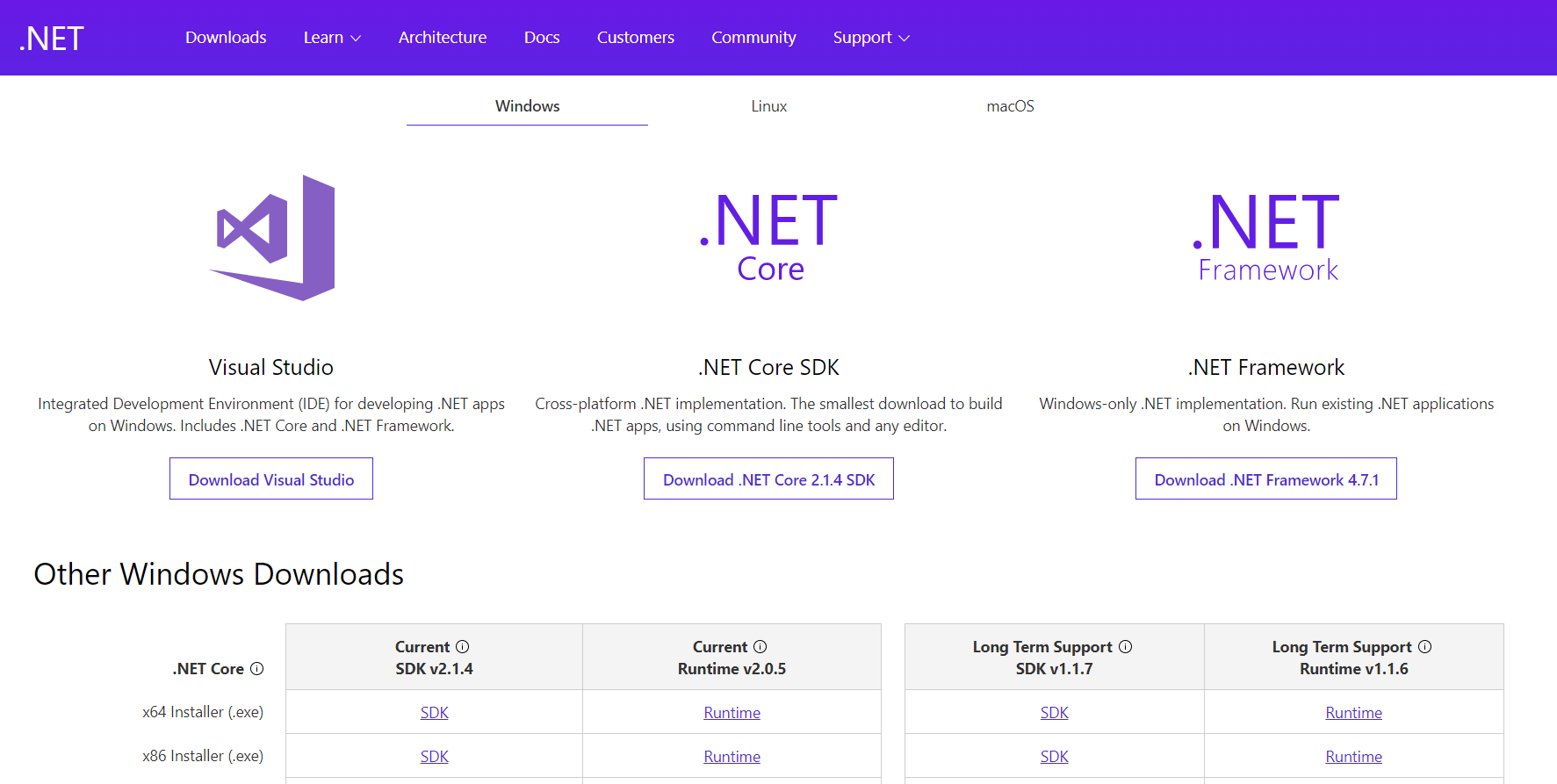
* The [Model-View-Controller (MVC) pattern](https://docs.microsoft.com/en-us/aspnet/core/mvc/overview) helps make your web APIs and web apps [testable](https://docs.microsoft.com/en-us/aspnet/core/testing/index).
* [Razor Pages](https://docs.microsoft.com/en-us/aspnet/core/mvc/razor-pages/index) (new in ASP.NET Core 2.0) is a page-based programming model that makes building web UI easier and more productive.
* [Razor markup](https://docs.microsoft.com/en-us/aspnet/core/mvc/views/razor) provides a productive syntax for [Razor Pages](https://docs.microsoft.com/en-us/aspnet/core/mvc/razor-pages/index) and [MVC views](https://docs.microsoft.com/en-us/aspnet/core/mvc/views/overview).
* [Tag Helpers](https://docs.microsoft.com/en-us/aspnet/core/mvc/views/tag-helpers/intro) enable server-side code to participate in creating and rendering HTML elements in Razor files.
* Built-in support for [multiple data formats and content negotiation](https://docs.microsoft.com/en-us/aspnet/core/mvc/models/formatting) lets your web APIs reach a broad range of clients, including browsers and mobile devices.
* [Model binding](https://docs.microsoft.com/en-us/aspnet/core/mvc/models/model-binding) automatically maps data from HTTP requests to action method parameters.
* [Model validation](https://docs.microsoft.com/en-us/aspnet/core/mvc/models/validation) automatically performs client- and server-side validation.

Client-side development

ASP.NET Core integrates seamlessly with popular client-side frameworks and libraries, including [Angular](https://docs.microsoft.com/en-us/aspnet/core/spa/angular), [React](https://docs.microsoft.com/en-us/aspnet/core/spa/react), and [Bootstrap](https://docs.microsoft.com/en-us/aspnet/core/client-side/bootstrap).

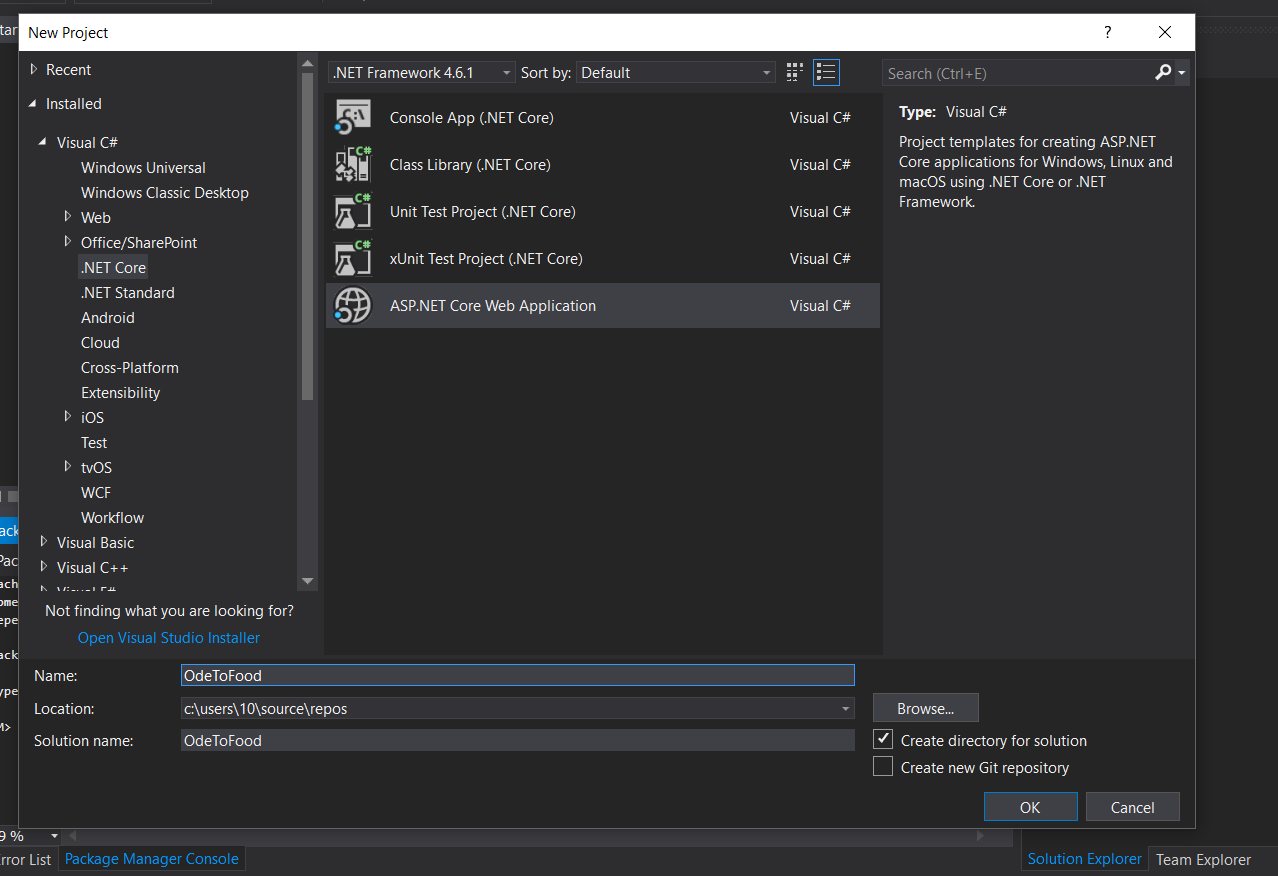
1. Building Your First ASP.NET Core Application
   1. Download and install .NET Core SDK:

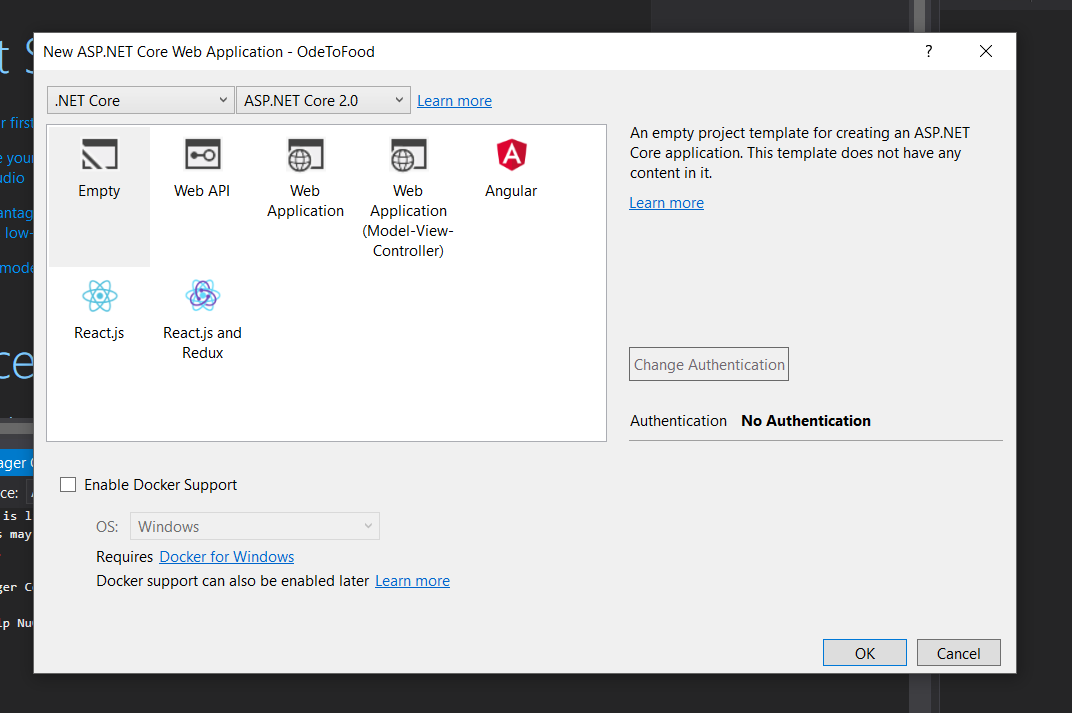
Download From: <https://www.microsoft.com/net/download/windows>



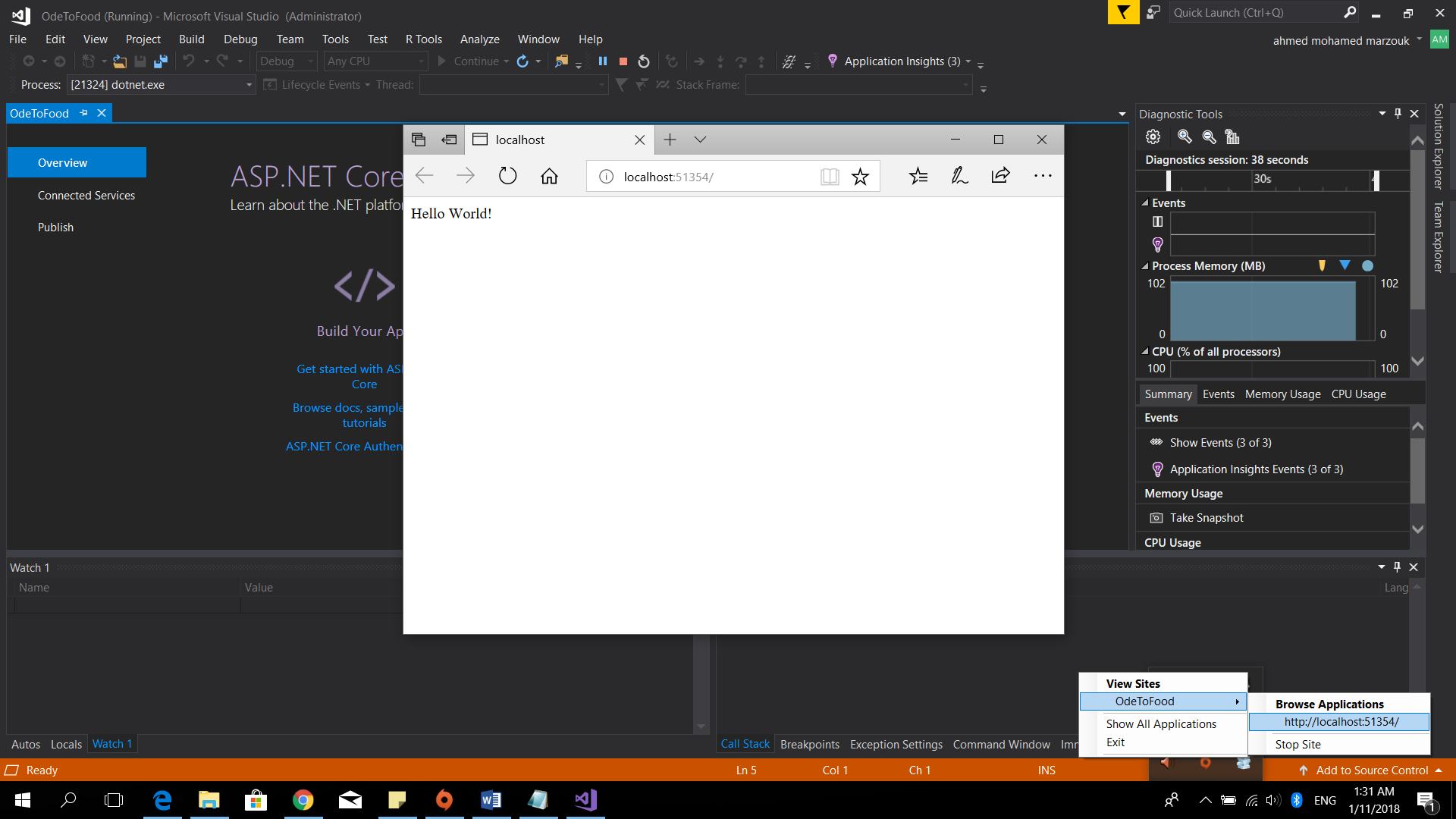
Then install.

* 1. A new project:





By default the web app run using IISExpress:



* 1. Command line tools and code: