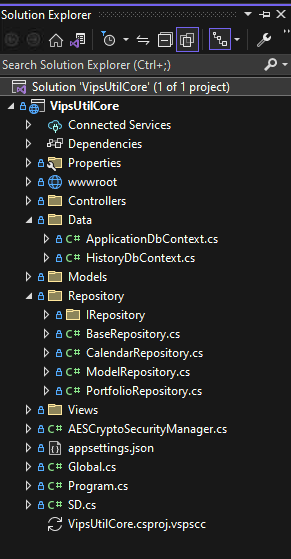
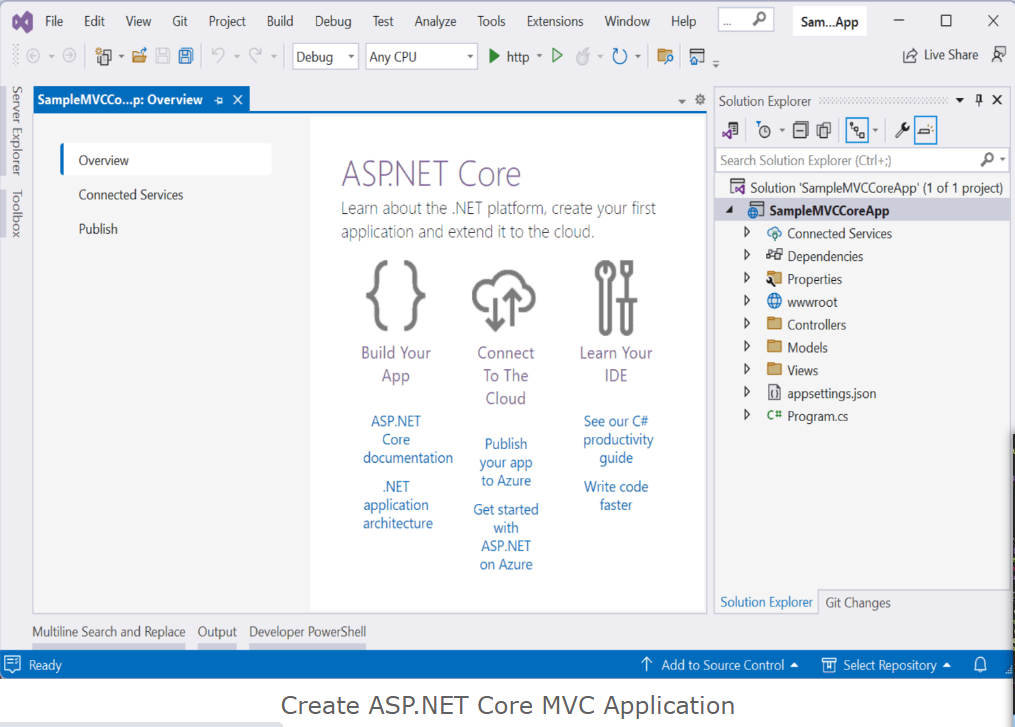
.Net Core Code level Understating

.Net 7

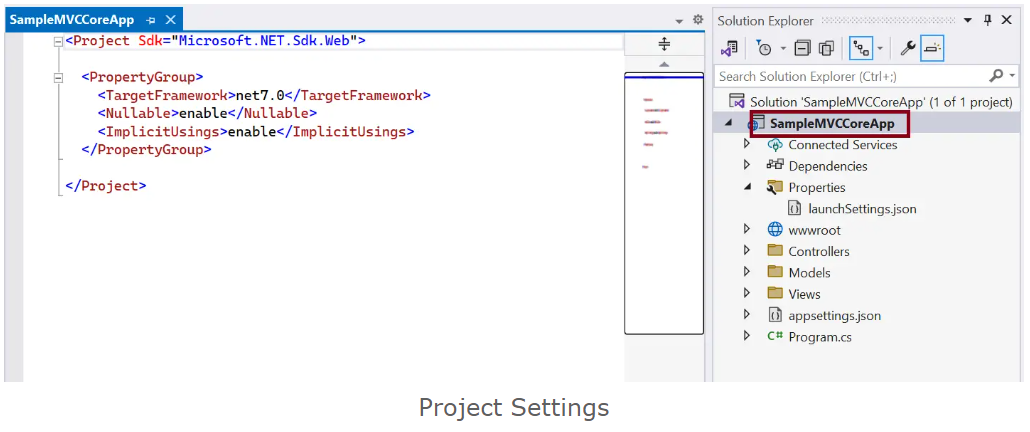


<https://www.tutorialsteacher.com/core/first-aspnet-core-application>



1. Connected Services

The 'Connected Services' node contains the **list of external services**, APIs, and other data sources. It helps in the integration with various service providers, such as Azure, AWS, Google Cloud, and third-party services like authentication providers or databases.



## Dependencies

The Dependencies node contains the **list of all the dependencies** that your project relies on, including NuGet packages, project references, and framework dependencies.

It contains two nodes, Analyzers and Frameworks.

**Analyzers** are **extensions for static code analysis**. They help you to enforce coding standards, identify code quality issues, and detect potential problems in your code. Analyzers can be custom rules or third-party analyzers provided by NuGet packages.

**Frameworks** node **contains the target framework** that your project is designed to run on. We have created an ASP.NET Core MVC application. So, it contains two frameworks, the .NET Core (Microsoft.NETCore.App) and ASP.NET Core (Microsoft.AspNetCore.App) framework. Click on any node and press F4 to see it's version, file path, etc.

Properties

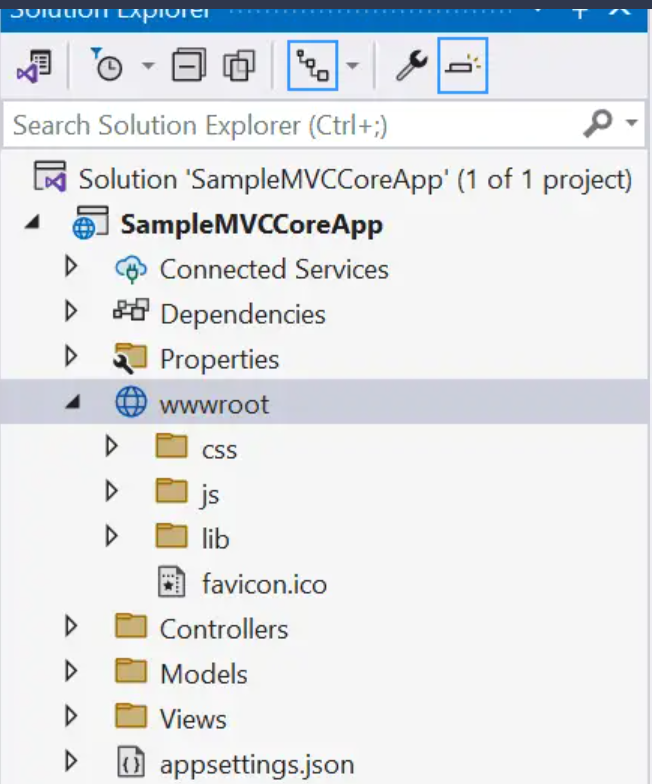
The Properties node includes launchSettings.json file which includes Visual Studio profiles of debug settings. launchSettings.json helps developers to configure the debugging and launch profiles of their ASP.NET (also known as ASP.NET Core) applications for different environments such as development, staging, production, etc.. The following is a default launchSettings.json file.



## wwwroot

By default, the **wwwroot** folder in the ASP.NET Core project is treated as a web root folder. Static files can be stored in any folder under the web root and accessed with a relative path to that root.

All the css, JavaScript, and external library files should be stored here which are being reference in the HTML file.

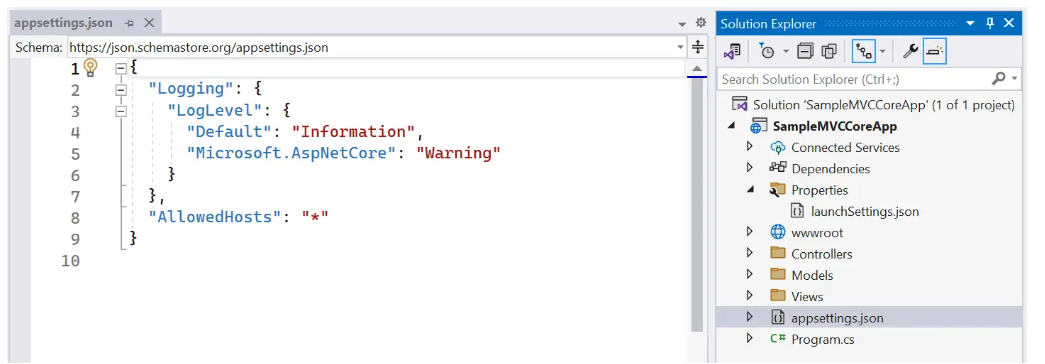


## Controllers, Models, Views

The Controllers, Models, and Views folders include controller classes, model classes and cshtml or vbhtml files respectively for MVC application.

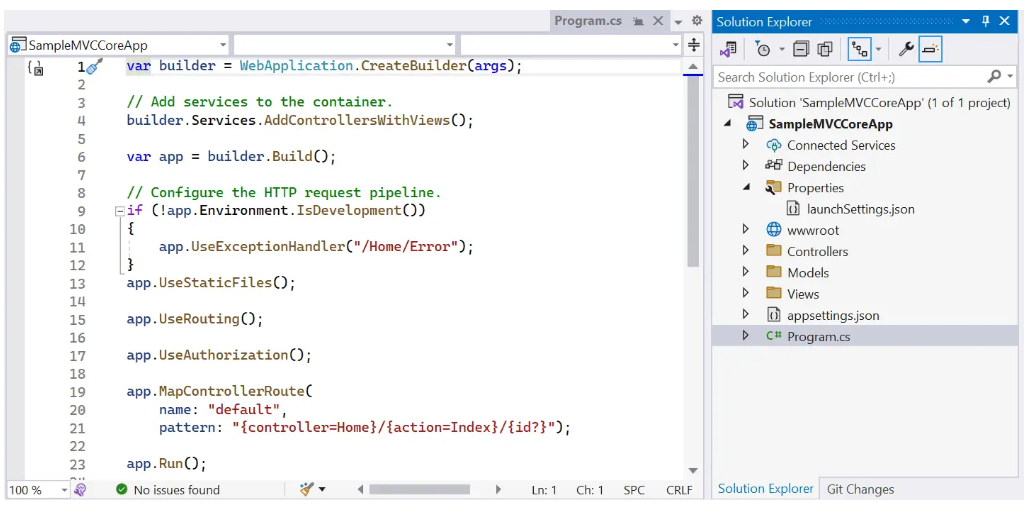
## appsettings.json

The appsettings.json file is a configuration file commonly used in .NET applications, including ASP.NET Core and ASP.NET 5/6, to store application-specific configuration settings and parameters. It allows developers to use JSON format for the configurations instead of code, which makes it easier to add or update settings without modifying the application's source code.



## program.cs

The last file 'program.cs' is an entry point of an application. ASP.NET Core web application is a console application that builds and launches a web application.



Every ASP.NET Core web applications starts like a console application then turn into web application. When you press F5 and run the application, it starts the executing code in the Program.cs file.

Notice that there is no Main() method in the above Progrm.cs file because it uses top-level statements feature of C# 9.

The first line creates an object of WebApplicationBuilder with preconfigured defaults using the CreateBuilder() method.

var builder = WebApplication.CreateBuilder(args);

The CreateBuilder() method setup the internal web server which is Kestrel. It also specifies the content root and read application settings file appsettings.json.

Using this builder object, you can configure various things for your web application, such as dependency injection, middleware, and hosting environment. You can pass additional configurations at runtime based on the runtime parameters.

The builder object has the Services() method which can be used to add services to the dependency injection container.

So, program.cs contains codes that sets up all necessary infrastructure for your application.