.NET Visual Studio Solution

.NET

*.NET **projects** are contained within a **solution**. A **solution** is a container for one or more related **projects**.*

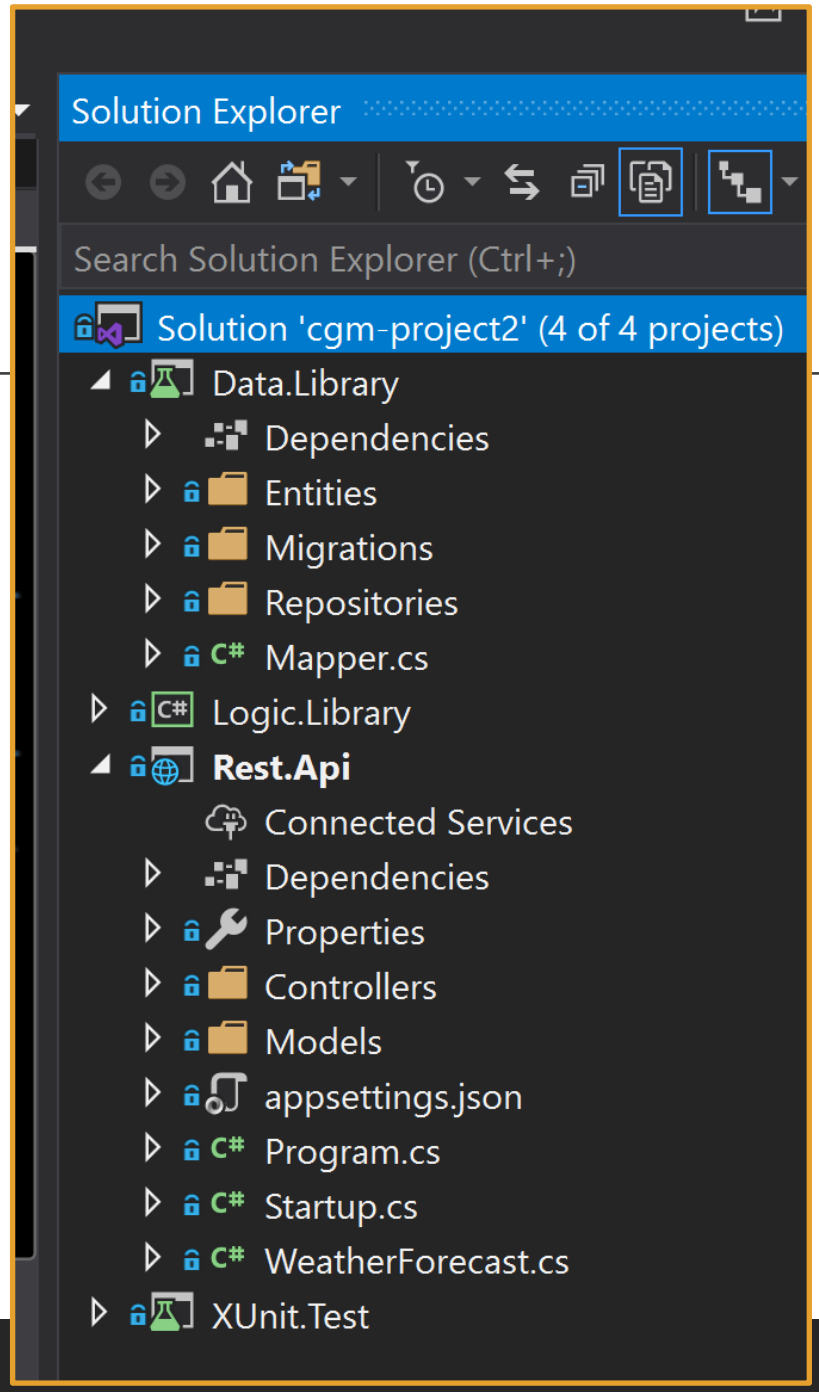
https://docs.microsoft.com/en-us/visualstudio/ide/solutions-and-projects-in-visual-studio?view=vs-2019

.NET Solution

https://docs.microsoft.com/en-us/visualstudio/ide/solutions-and-projects-in-visual-studio?view=vs-2019#solutions

A ***Solution*** is a container for one or more related projects along with their build information, Visual Studio window settings, and any miscellaneous files that aren't associated with one particular project.

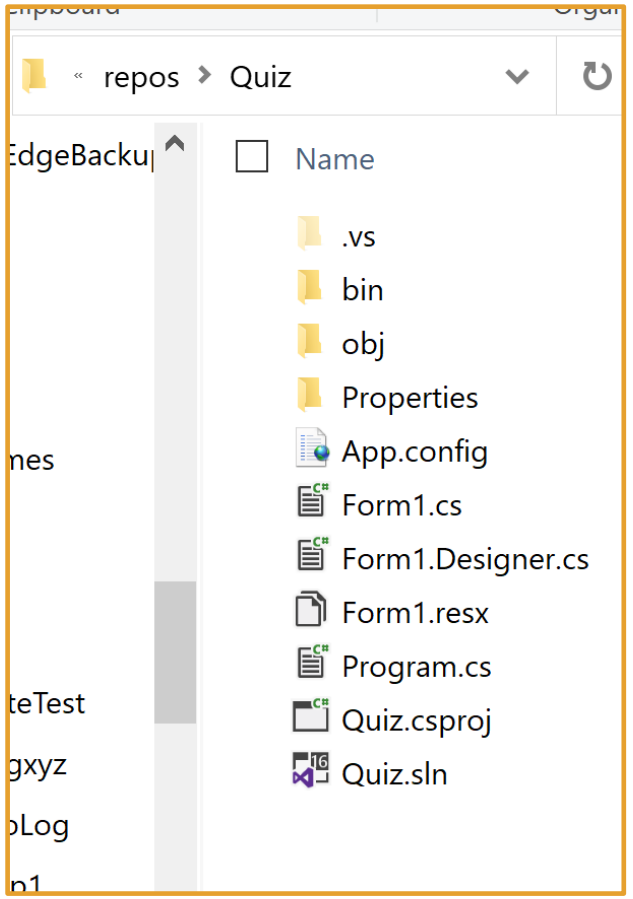
A ***solution*** is described by a text file (extension .sln) in XML format. It is not intended to be edited by hand.



.NET Solution - Projects

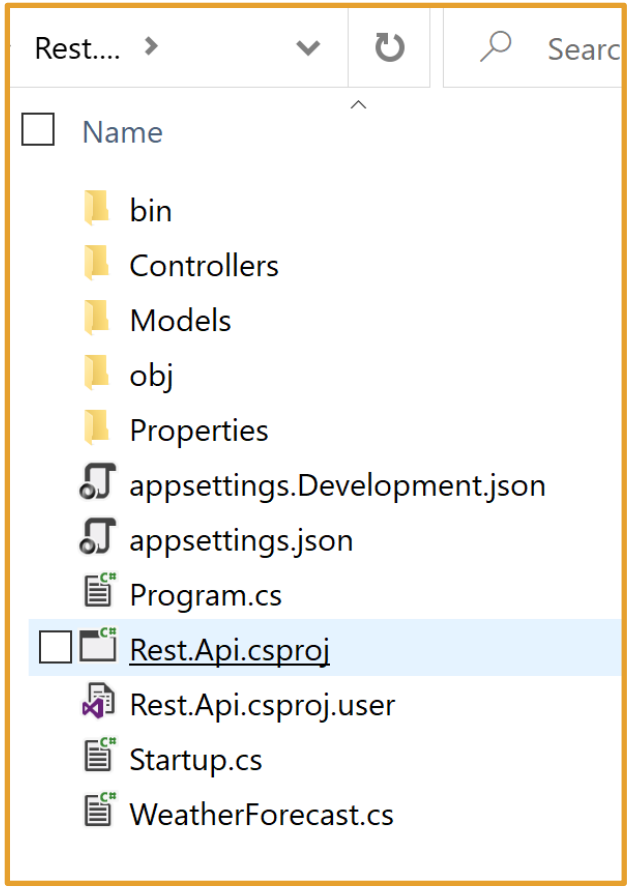
https://docs.microsoft.com/en-us/visualstudio/ide/solutions-and-projects-in-visual-studio?view=vs-2019
https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/program-structure

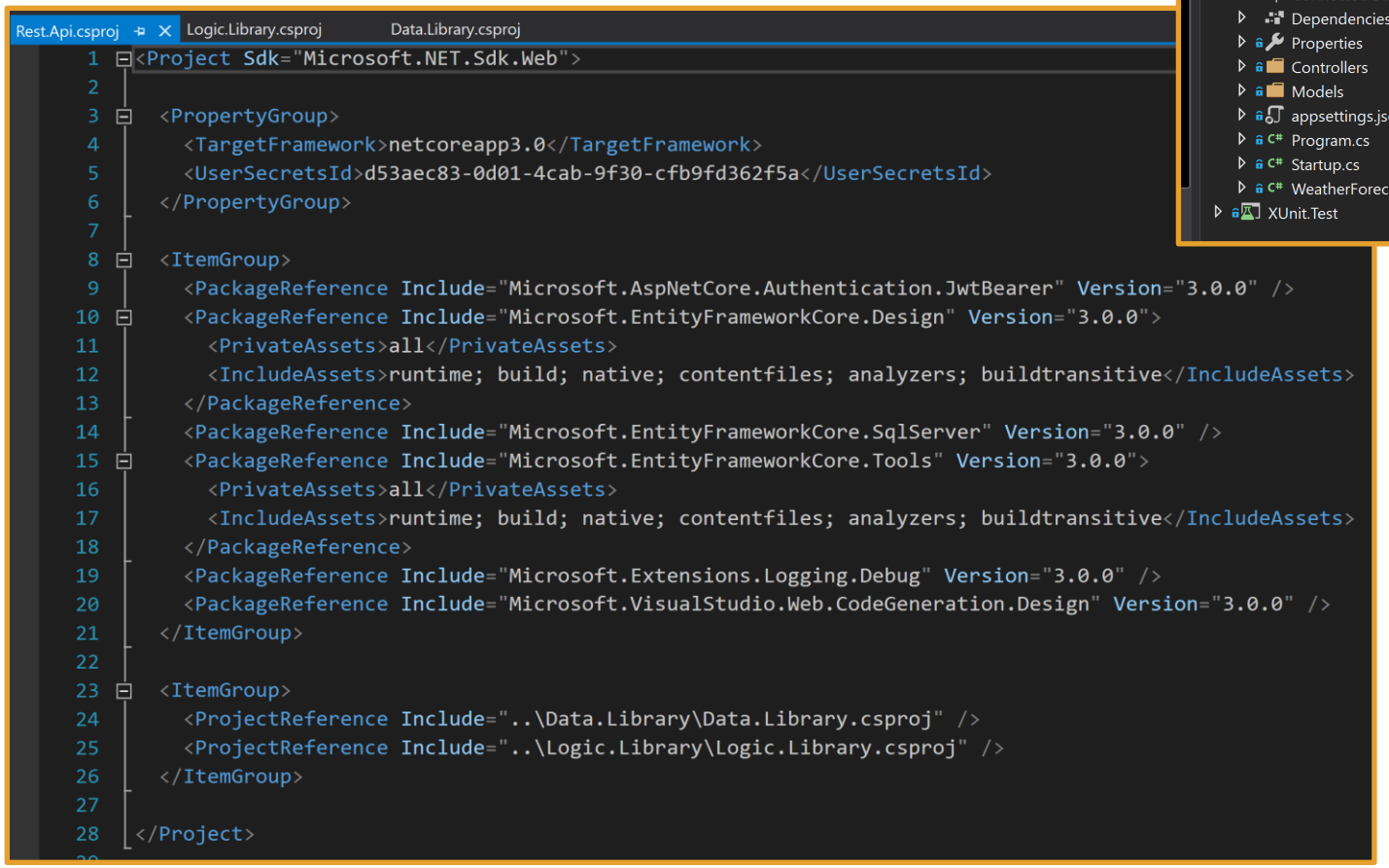
- An app in Visual Studio starts with a project. A project contains all files that, when compiled, are placed into an executable (.exe) or a library (.dll).
- Files can include source code, icons, images, data files, etc.
- A project contains compiler settings and other configuration files that might be needed by various services or components.
- Visual Studio uses ***MSBuild*** to build each project in a solution, and each project contains an ***MSBuild*** project file.
- The file extension for a C# project is .csproj.
- The project file is an ***XML*** document that contains all the information and instructions that ***MSBuild*** needs in order to build a project including the content, platform requirements, versioning information, web server or database server settings, and the tasks to perform.

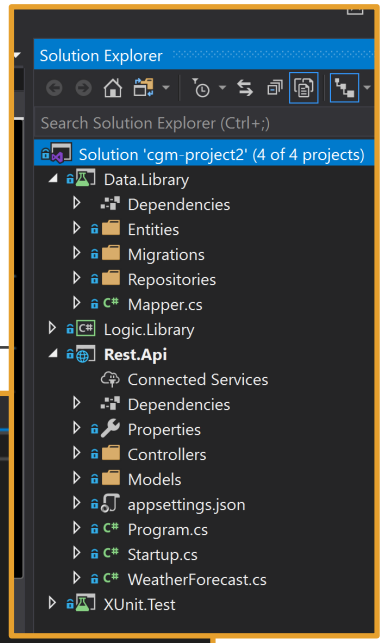


.NET Solution - Projects

https://docs.microsoft.com/en-us/visualstudio/ide/solutions-and-projects-in-visual-studio?view=vs-2019
https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/program-structure







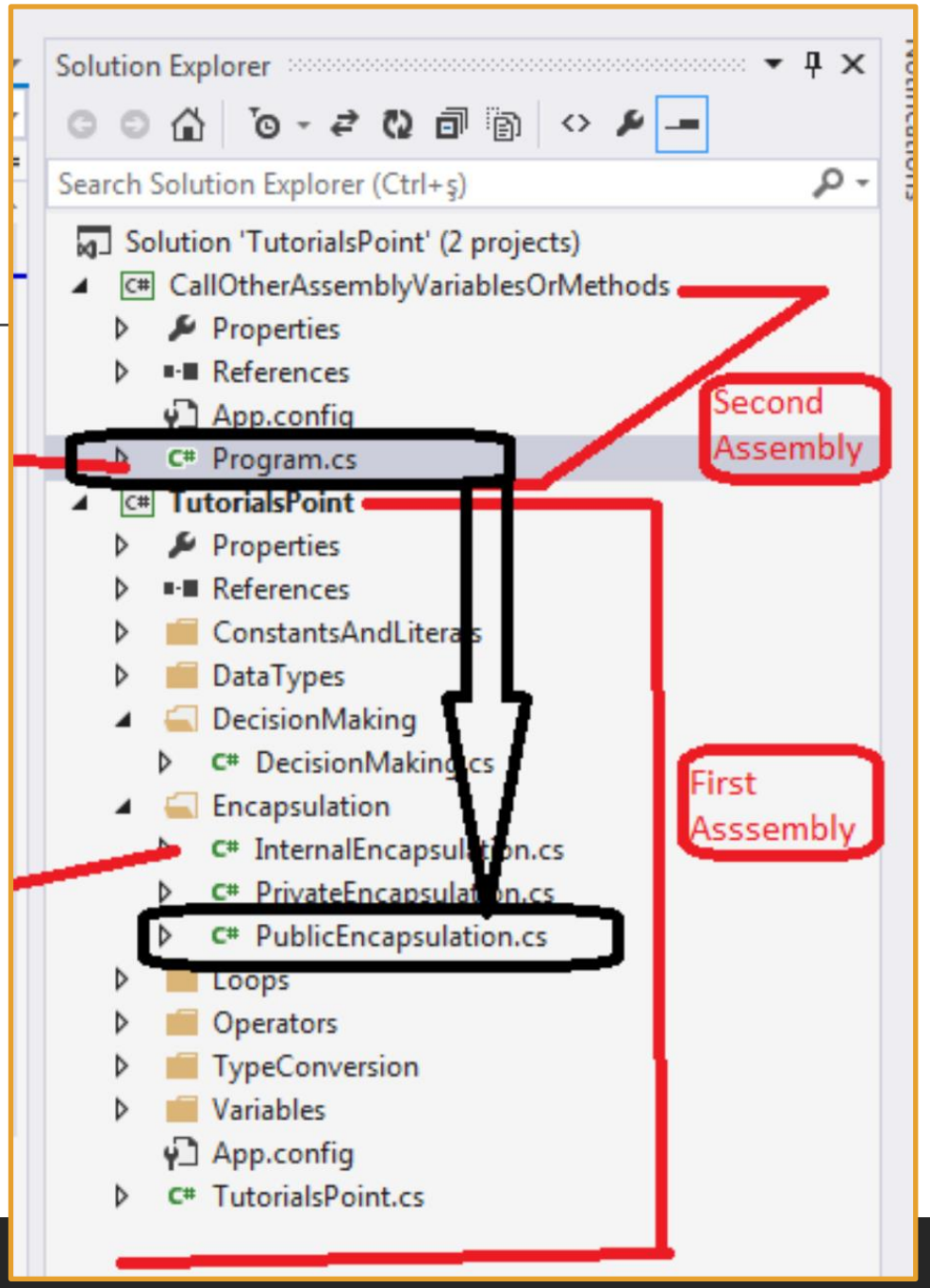
.NET Solution - Assembly

https://docs.microsoft.com/en-us/dotnet/standard/assembly/

Assemblies:

- form the fundamental units of deployment, version control, reuse, activation scoping, and security permissions for ***.NET***-based applications.
- are a collection of types and resources that work together and form a logical unit of functionality.
- take the form of ***executable (.exe)*** or ***dynamic link library (.dll)*** files.
- provide the ***Common Language Runtime*** with the information it needs to be aware of ***type*** implementations.

In ***.NET***, you can build an assembly from one or more source code files. Each projects' files are compiled (combined) into one .dll or .exe file called an ***Assembly***.



.NET Solution - Assembly

https://docs.microsoft.com/en-us/dotnet/standard/assembly/

An ***assembly*** is:

- Code that the CLR executes. Each ***assembly*** can have only one entry point (Main).
- Security boundary. An ***assembly*** is the unit at which permissions are requested and granted.
- Version boundary. The assembly is the smallest versionable unit in the CLR. All ***types*** and resources in the same ***assembly*** are versioned as a unit.
- Deployment unit. When an application starts, only the ***assemblies*** that the application initially calls must be present. Other ***assemblies*** are retrieved on demand. This is called ***Just-In-Time (JIT) compiling***.