# Project Setup

Follow **net-commands.txt** file in the root to setup the project and also for the dotnet general commands

The following pic shows the folder structure:

A screenshot of a computer program

Description automatically generated

# Updating

## Application URLs

Go to Properties 🡺launhsettings.json

### Remove section

,

    "https": {

      "commandName": "Project",

      "dotnetRunMessages": true,

      "launchBrowser": true,

      "launchUrl": "swagger",

      "applicationUrl": "https://localhost:5217;http://localhost:5218",

      "environmentVariables": {

        "ASPNETCORE\_ENVIRONMENT": "Development"

      }

    }

### Update section

A computer screen with text and images

Description automatically generated

## Appsettings.development.json

A screen shot of a computer code

Description automatically generated

# Running the Application

* Navigate to the OAuth2.WebApi folder
* Issue command
  + dotnet build
  + dotnet run or dotnet watch run
  + User CTRL + C to stop
* Then go to <https://localhost:5217/swagger> or <http://localhost:5218/swagger> to execute

If problem running https then run command prompt as an administrator and issue

* dotnet dev-certs https --clean
* dotnet dev-certs https --trust

# VS Code Extensions

Install following extensions

1. C# Dev Kit – Microsoft
   1. This will also bring in C# - Base language support for C#
   2. .net Install tool for extension authors – Microsoft
   3. IntelliCode for C# Dev Kit - Microsoft
2. Material icon theme – Philip Kief
3. C# Extensions – JosKreativ [not needed due to C# Dev Kit]
4. NuGet Gallery v0.0.24
5. SQLite - alexcvzz

# VS Code Troubleshooting

When intelisence is not working or other issues then press CTRL+SHIFT+P and type Developer: reload window

# Disabling Warning for Non-nullable Property

To disable the following warning

A screenshot of a computer program

Description automatically generated

Open the project file OAuth2.WebApo.csproj and change

|  |  |
| --- | --- |
| From | To |
|  |  |

# Disable ImplicitUsings

## Impact

When ImplicitUsing is change to “disable” in the OAuth2.Webapi.csproj file then program.cs file will start throwing error(s).

|  |  |
| --- | --- |
| From | To |
|  |  |

You’ll need to add the using statement to fix it.

A screen shot of a computer

Description automatically generated

So here is the updated program.cs file code.

A screen shot of a computer program

Description automatically generated

## What to do?

I have enabled ImplicitUsings in my code to make it easier to build items.

You decide for yourself.

# Entity Framework (EF)

1. An Object Relational Mapper (ORM)
2. Translates our code into SQL commands that update our tables in the database
3. We get a DbContext class with EF
   1. Create a class that derives from DbContext class
   2. This class acts as a bridge between the entity classes and database
4. EF works with database provide. For this project will work with SQLite, for development only
   1. SQLite Provider or Database Provider is responsible for converting LINQ queries to SQL commands

A black and white diagram

Description automatically generated

## EF Features

* Querying
* Change Tracking
* Saving
* Concurrency
* Transactions
* Caching
* Built-in conventions
* Configurations
* Migrations

# Adding Packages

NuGet Gallery v0.0.24 is already installed [above](#_VS_Code_Extensions).

## Open NuGet Gallery

Press CTRL+SHFT+P to go to command pallet

A black and blue rectangle with blue lines

Description automatically generated

## Microsoft.EntityFrameworkCore.Sqlite

Search for it (do not pick Microsoft.EntityFrameworkCore.Sqlite.Core).

Select the project and then click install

A screen shot of a computer

Description automatically generated

## Microsoft.EntityFrameworkCore.Design

Select the project and then click install

## System.IdentityModel.Tokens.Jwt

Search for System.IdentityModel.Token.Jwt by Microsoft

Select the project and then click install

A screenshot of a computer

Description automatically generated

## Microsoft.AspNetCore.Authentication.JwtBearer

Search for Microsoft.AspNetCore.Authentication.JwtBearer by Microsoft

Select the project and then click install

A screenshot of a black screen

Description automatically generated

## dotnet-ef

This package is not available on NuGet Gallery so search for it and then pick the command.

<https://www.nuget.org/packages/dotnet-ef/>

> dotnet tool install --global dotnet-ef --version 7.0.10

If you get into an error regarding tool already installed then check it

> dotnet tool list -g

Package Id Version Commands

--------------------------------------

dotnet-ef 6.0.6 dotnet-ef

**Update Tool**

> dotnet tool update --global dotnet-ef --version 7.0.10

Tool 'dotnet-ef' was successfully updated from version '6.0.6' to version '7.0.10'.

Commands related to dotnet tool: <https://learn.microsoft.com/en-us/dotnet/core/tools/dotnet-tool-install>

# EditorConfig

Create a file .editorconfig in the root and add following content into it.

[\*.{cs,vb}]

dotnet\_naming\_rule.private\_members\_with\_underscore.symbols  = private\_fields

dotnet\_naming\_rule.private\_members\_with\_underscore.style    = prefix\_underscore

dotnet\_naming\_rule.private\_members\_with\_underscore.severity = suggestion

dotnet\_naming\_symbols.private\_fields.applicable\_kinds           = field

dotnet\_naming\_symbols.private\_fields.applicable\_accessibilities = private\_fields

dotnet\_naming\_style.prefix\_underscore.capitalization    = camel\_case

dotnet\_naming\_style.prefix\_underscore.required\_prefix   = \_

A screenshot of a computer

Description automatically generated

# Adding GlobalJson

Add globaljson file in the root by issuing command

> dotnet new globaljson

It will tell which version of the api to use

A screenshot of a computer

Description automatically generated

# Disabling Hot Reload

>dotnet watch run

* Makes hot reload enabled.
* There are time where this doesn’t work

It could be turned off by updating Properties/launchSettings.json and updating with following key.

A screen shot of a computer

Description automatically generated

Sometimes this may not work as well. Better option would be

>dotnet watch –no-hot-reload