Table of Contents

[App Basics 2](#_Toc106832390)

[.Net 6 Install 2](#_Toc106832391)

[VisualStudio Code Install 2](#_Toc106832392)

[nodeJs Install 2](#_Toc106832393)

[Install Postman 2](#_Toc106832394)

[Angular Install 2](#_Toc106832395)

[Adding c# related extensions in VS Code 3](#_Toc106832396)

[Angular Extensions 3](#_Toc106832397)

[Making VS Code IDE Better 3](#_Toc106832398)

[WebApi Project 4](#_Toc106832399)

[Setting up .Net API Project 4](#_Toc106832400)

[Create WebAPI Method 1 4](#_Toc106832401)

[Create WebAPI Method 2 4](#_Toc106832402)

[Tidy up the Solution 4](#_Toc106832403)

[Actual Commands used for creating the api 4](#_Toc106832404)

[Folder Structure 4](#_Toc106832405)

[Running .Net API Project 4](#_Toc106832406)

[Switching to Classic hosting model 5](#_Toc106832407)

[Implicit Using Statement 5](#_Toc106832408)

[Program.cs 5](#_Toc106832409)

[WeatherForecast.cs 5](#_Toc106832410)

[/Controller/WeatherForecastController 5](#_Toc106832411)

[Angular Project 6](#_Toc106832412)

[Setting up Angular App 6](#_Toc106832413)

[Create the Anuglar app 6](#_Toc106832414)

[Add resources bootstrap and font-awsome 6](#_Toc106832415)

[App Features 6](#_Toc106832416)

[What we will be using? 6](#_Toc106832417)

[Important Documents 6](#_Toc106832418)

# App Basics

|  |  |
| --- | --- |
| .Net 6 Install  1. Got to dotnet.microsoft.com/download 2. Click download 3. Select your OS Windows/Linux/macOS/Docker 4. Install SDK x64/x86  VisualStudio Code Install We’ll use VS Code as the code editor: <https://code.visualstudio.com/> | nodeJs Install  1. Go to <https://nodejs.org/en/> 2. Then other downloads 3. Select your OS 4. Either select the current version or select Node.js 16.13.0 from previous releases 5. Look into installing different versions of nodejs with NVM: <https://joachim8675309.medium.com/installing-node-js-with-nvm-4dc469c977d9> |
| Install Postman <https://www.postman.com/downloads/>  No need to create an account to use Postman. Towards the bottom there is skip link to skip login | Angular Install Installing it globally. Nodejs must be installed first   * npm uninstall -g @angular/cli * npm cache clean * npm install -g @angular/cli@13.0.2 |

|  |  |
| --- | --- |
| > dotnet --list-sdks | >dotnet --info |
| >node --version  v16.13.0 |
| >npm --version  8.1.3  **Alternate use nvm to install different versions of node**  <https://joachim8675309.medium.com/installing-node-js-with-nvm-4dc469c977d9> |
| >ng –version |

# Adding c# related extensions in VS Code

|  |  |
| --- | --- |
| 1. C# for Visual Studio Code (powered by OmniSharp) 2. C# Extensions by JosKreativ 3. Material Icon Theme by Philipp Kief     adding nuget   1. vscode-nuget-package-manager | * After the reload you'll be shown some file missing popup - click yes. It will create .vscode folder. * If you miss this then do CTRL+SHIFT+P and type assets and click it to adding missing assets |

# Angular Extensions

|  |  |
| --- | --- |
| * Angular and then select Angular v7 Snippets by john papa * Angular Files 1.6.2 Alexander Ivanichev * Angular Language Service 0.1.10 by Angular * Angular2-switcher by infinity1207 * Auto Rename Tag 0.0.15 Jun Han | * Bracket Pair Colorizer CoenraadS * Debugger for Chrome Microsoft 4.11.0 * Material Icon Theme Philipp Kief 3.6.0 * Path Intellisense Christian Kohler 1.4.2 * Prettier - Code formatter Esben Petersen 1.6.1 * TSLint egamma 1.4.40 |

# Making VS Code IDE Better

1. AutoSave: Go to File and select “AutoSave” to automatically save our changes
2. Go to File > Preferences > Settings
   1. Type Font and change
      1. Main font size
      2. Scroll down and change for Console and Terminal as well
   2. Type exclude and add following to hide BIN and OBJ folders
      1. \*\*/bin
      2. \*\*/obj
   3. Type folders, go to Explorer:Compat Folder and unselect

# WebApi Project

Create a base folder to house the projects and files

## Setting up .Net API Project

|  |  |
| --- | --- |
| Create WebAPI Method 1 >dotnet new sln  Solution name the same as the container folder  > dotnet new sln --name MySolution  Solution name with custom name  >dotnet new webapi -o MSC.WebApi  Create a new project with name MCS.WebApi  >dotnet sln add MSC.WebApi  Add the project to the solution Create WebAPI Method 2 >dotnet new webapi -o MSC.WebApi -n MCS.WebApi Tidy up the Solution File > Preference > Settings and then type “Exclude”  Exclude Bin and obj folder by following the examples | Actual Commands used for creating the api Create a dir to house web api solution and project  >md MySocialConnect-API  cd into new dir  >cd MySocialConnect-API  Create a new solution  >dotnet new sln --name MSC-API  Create a new WebApi project  >dotnet new webapi -o MSC.Api  Add the project to the solution  >dotnet sln add MSC.Api Folder Structure |

## Running .Net API Project

Open command prompt and navigate to “[basePath]/MySocialConnect-API/MSC.Api”

Then execute “dotnet run” or “dotnet run watch” commands.

Once running successfully then pick the url from the command prompt

Text

Description automatically generated

Then go to <https://localhost:7135/swagger> or <http://localhost:5157/swagger>

Swagger will display. Expand the GET method end point under WeatherForecast, click Try it out and then click Execute. You should see result here.

## Switching to Classic hosting model

Go to folder Documents/dotnet/ClassicHostingModel

There are two files, Program.cs and Startup.cs.

Put these in the MSC.Api folder. Startup.cs is a new file and Program.cs will get replace.

Also go to MSC.Api/Properties and open launchSettings.json file. Change the launchUrl. For this following this link [Tutorial: Create a web API with ASP.NET Core | Microsoft Docs](https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-6.0&tabs=visual-studio-code)

## Implicit Using Statement

In Program.cs, WeatherForecast.cs and /Controller/WeatherForecastController there are using statement missing. This is due to a flag in MSC.Api.csproj file. Comment out the ImplicitUsings.

Text

Description automatically generated

Now when you go to Program.cs, WeatherForecast.cs and /Controller/WeatherForecastControoler you’ll see a lot of error since the using statement is missing.

Put the cursor on each error and then click CTRL+. to use the using statement.

### Program.cs

Following will get added

* using Microsoft.AspNetCore.Builder;
* using Microsoft.Extensions.DependencyInjection;
* using Microsoft.Extensions.Hosting;

### WeatherForecast.cs

Following will get added

* using System;

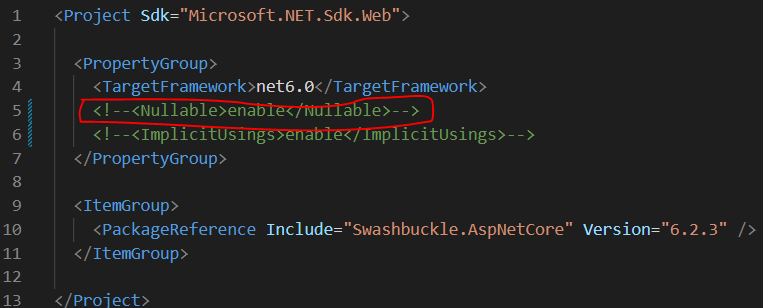
### /Controller/WeatherForecastController

Following will get added

* using System;
* using System.Collections.Generic;
* using System.Linq;
* using Microsoft.AspNetCore.Mvc;
* using Microsoft.Extensions.Logging;

## Nullable Enabled

If you look at the WeatherForecast.cs, the stril property Summary is nullable. This will cause some issues for us so we will remove the ? from it and also comment out the nullable flag from MSC.Api.csproj file.



And then from WeatherForecast.cs remove ?



# Angular Project

## Setting up Angular App

|  |  |
| --- | --- |
| Create the Anuglar app >ng new MySocialConnect-SPA  >ng serve to run Add resources bootstrap and font-awsome >npm install bootstrap font-awesome  go to angular.json, look at the Styles Array. it will be referencing "src/styles.css". Open this css file and import the bootstrap and font-awesome   * @import '../node\_modules/bootstrap/dist/css/bootstrap.min.css';   @import '../node\_modules/font-awesome/css/font-awesome.min.css'; |  |

|  |  |
| --- | --- |
| App Features  1. Registration & Login with ASP.Net Identity 2. View list of members currently online 3. Like members and list 4. View members who liked them 5. Upload photo 6. Update member profile 7. Messaging (live chat) system to message member’s real time 8. Pagination 9. Caching | What we will be using?  * Entity Framework * HTML5 * Bootstrap * CSS * TypeScript * C# * NoSQL |

# Important Documents

1. [0002 VS Code DotNet Angular Commands.docx](0002%20VS%20Code%20DotNet%20Angular%20Commands.docx)
2. [0003 Working with Sample WeatherForecastController.docx](0003%20Working%20with%20Sample%20WeatherForecastController.docx)