Table of Contents

[VS Code 2](#_Toc113051062)

[Commands 2](#_Toc113051063)

[Adding c# related extensions in VS Code 2](#_Toc113051064)

[Making VS Code IDE Better 2](#_Toc113051065)

[DotNet Commands 3](#_Toc113051066)

[Create WebAPI Method 1 3](#_Toc113051067)

[Create WebAPI Method 2 3](#_Toc113051068)

[Tidy up the Solution 3](#_Toc113051069)

[Run and Debug 3](#_Toc113051070)

[Angular Commands 4](#_Toc113051071)

[Commands 4](#_Toc113051072)

# VS Code

## Commands

|  |  |
| --- | --- |
| CTRL+K U Ctrl+K W => close all window  CTRL+K S to save all  CTRL+/ to comment the line  Shift+Alt A to comment a block  CTRL+SHIFT P show all commands  F5 Debug  CTRL+F5 start without debugging  CTRL+` to toggle the terminal  CTRL+. to bring up the context menu on squigly lines | >cls => to clear the screen  >dotnet --version  >node --version  >npm --version  >dotnet -h => see the commands  >dotnet new -h => contextual help  >dotnet new webapi -h => contextual help |

Terminal>dotnet run will run the project

Terminal>dotnet watch run will re-run after save

CTRL+C to shutdown

## 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 |

## 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

# DotNet Commands

## Create WebAPI Method 1

>dotnet new sln

Solution name the same as the container folder

>dotnet new webapi -o MSC.WebApi

Create a new project with name MSC.WebApi

>dotnet sln add MSC.WebApi

Add the project to the solution

## Create WebAPI Method 2

>dotnet new webapi -o MSC.WebApi -n MSC.WebApi

## Tidy up the Solution

File > Preference > Settings and then type “Exclude”

Exclude Bin and obj folder by following the examples

* Dotnet build
* dotnet run
* dotnet watch run
* dotnet clean
* dotnet build --no-incremental [rebuild]
* dotnet new gitignore

## Run and Debug

|  |  |
| --- | --- |
| Press CTRL+SHIFT+D  Click create a launch.json file  Select Net5+ and Net Core | Graphical user interface, application  Description automatically generated |

# Angular Commands

## Commands

**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';

**ngx-toastr**

Different versions of ngx-toastr

<https://www.npmjs.com/package/ngx-toastr?activeTab=versions>

This project is angular version 13.0.2 so latest will not work and following will throw an error

>npm install ngx-toastr

Instead will need to install an older version that support angular 13.0.2

>npm install [ngx-toastr@14.3.0](mailto:ngx-toastr@14.3.0)

Then go to angular.json and add the toastr.css to the styles array

* ./node\_modules/ngx-toastr/toastr.css

Go to app.modules.ts and add the ToastrModule to the imports array as well. Make sure to import it as well.

import { ToastrModule } from 'ngx-toastr';

And then to import array add

ToastrModule.forRoot({positionClass: 'toast-bottom-right'})

|  |  |
| --- | --- |
| * ng g c value -- skip-tests [component] * ng g c site/nav --skip-tests [component] * ng g s core/services/account --skip-tests [service] | * ng g interface Employee --type=model [interface] * ng g class EmployeeClass --type=model [class] |

to move between the component and html use shortcut key since we have the extension

* html=alt+o >component=alt+u
* in html mode type a- it will give the snippet to pick

**Decorators**

1. Input: move data from parent to child
2. Output: move data from child to parent

**Directives**

1. \*ngIf: structural directive – conditional content