1. Create new Project with dotnet CLI.

* Create directory for project work space.
* cd to directory and Run command CLI (Command Line Interface).

dotnet new angular

1. Wait for create directory for project.

* Font-end with Angular (Must have angular cli match version on dotnet created).
* Back-end with .net core.

1. Install default dependencies in font-end with **npm install –save** 0r **yarn install**

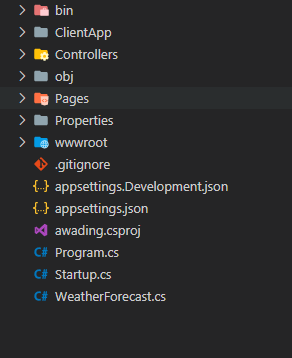
**\*\*\*** cd into **ClientApp** directory before run install dependencies command. \*\*\*

1. Can run the project with CLI “dotnet watch run” or “dotnet run”.
2. Develop and debug with logging on CMD (Log on Back-end (.net core) will display here).
3. See the web page on <https://localhost:5001>

* Font-end (Angular) code inside **ClientApp** directory.
* Other are Back-end code.

\*\* If Open editor with Visual Studio Code Extension must have this below.

* Angular Essentials
* C#

It will help to develop and read programming code with editor.

Project structure with [dotnet new]

After run sometime you can’t view web page on IE browser because it not support on IE. Then we can set the project to support or can run on IE in step below.

1. Open to **ClientApp** directory and uncomment line of **IE 9-11** in file **browserslist**

(Remove # out of this line).

1. Open file **src/polyfills.ts**

* Uncomment line **import 'classlist.js';**
* Uncomment line **import '** **web-animations-js';**
* Open CMD and cd to ClientApp directory and Run command
  + **npm install --save classlist.js** or **yarn install classlist.js**
  + **npm install --save web-animations-js** or **yarn install web-animations-js**

1. Open file **/ClientApp/tsconfig.json** Change target to **es5** like this bellow.
2. Re-run Project with CLI “dotnet watch run” or “dotnet run”.