# SOUTHEASTERN
## LOUISIANA UNIVERSITY

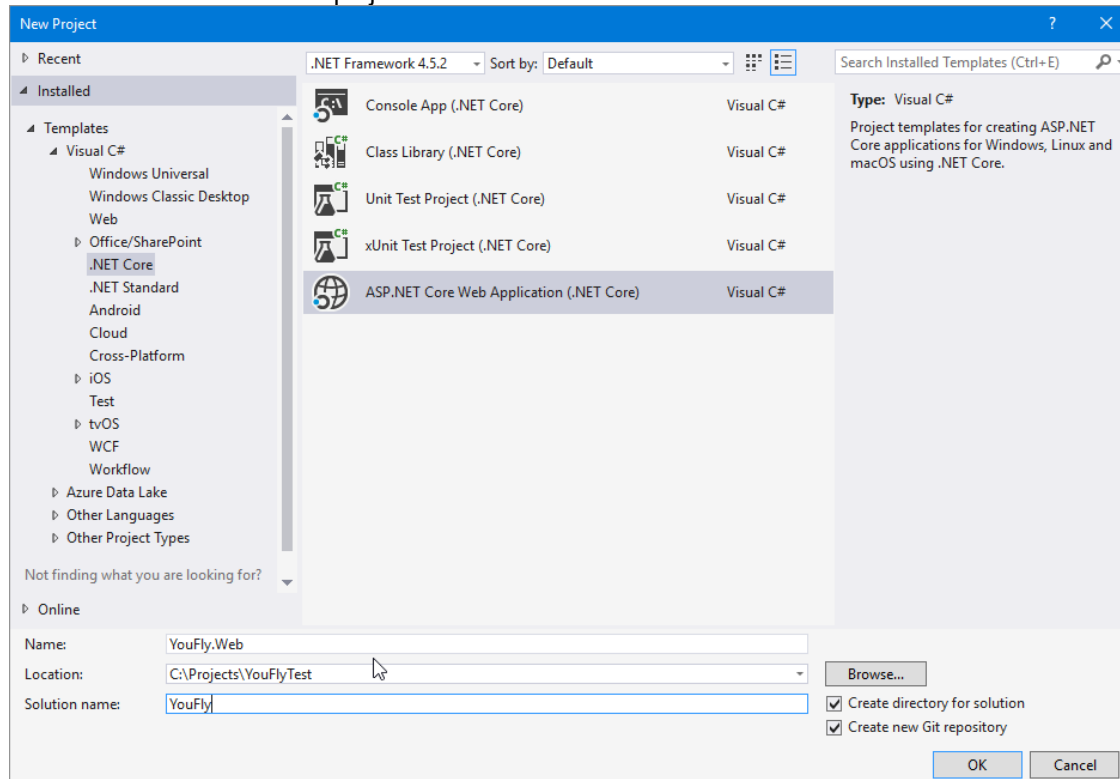Andrew Ilgenfritz <andrew.ilgenfritz@selu.edu>

## .NET Core and Angular 2 playing nice
8 messages

**Matthew Vidacovich** <mvid@envoc.com>
To: Andrew Ilgenfritz <andrew.ilgenfritz@selu.edu>, 383 <383@envoc.com>

Wed, Mar 15, 2017 at 3:25 PM

**There are several ways to attack this problem, this is one of the "uses less magic" ways of handling it.**

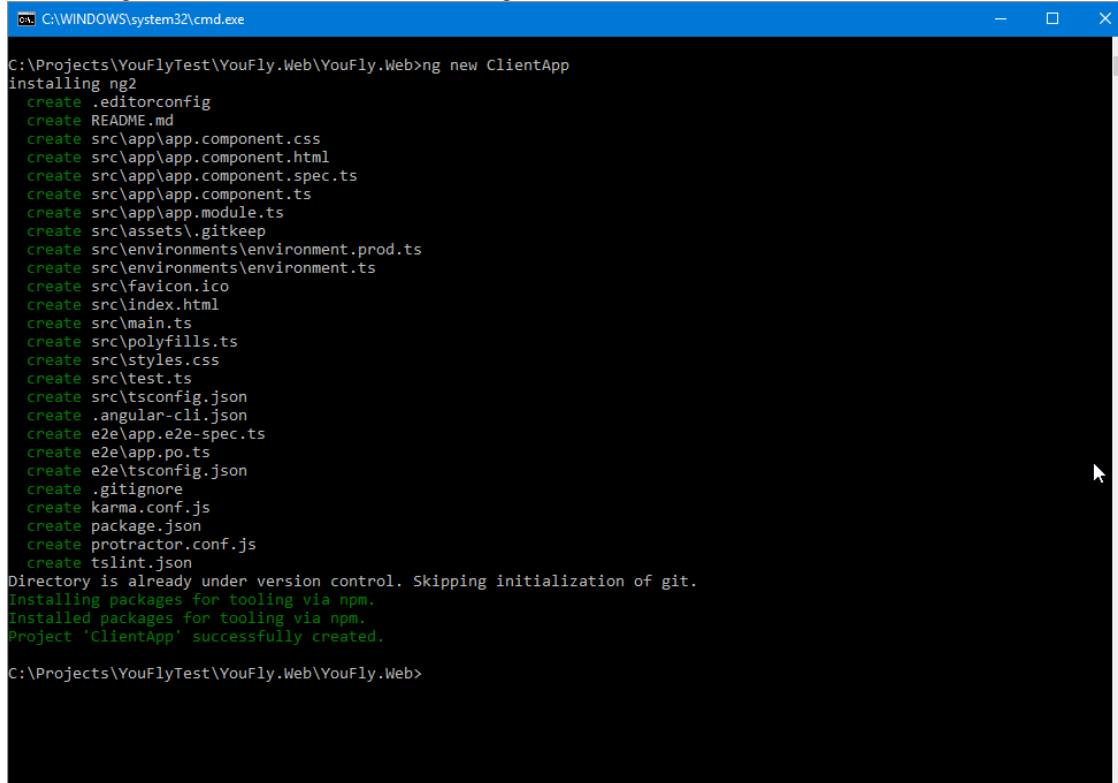I would make the NET core project:



(I like the "create new git repository" option)

Then in this folder:

Inline image 2

Run the ng new command to create the angular 2 instance:

```
C:\WINDOWS\system32\cmd.exe                                         —    □    ×

C:\Projects\YouFlyTest\YouFly.Web\YouFly.Web>ng new ClientApp
installing ng2
  create .editorconfig
  create README.md
  create src\app\app.component.css
  create src\app\app.component.html
  create src\app\app.component.spec.ts
  create src\app\app.component.ts
  create src\app\app.module.ts
  create src\assets\.gitkeep
  create src\environments\environment.prod.ts
  create src\environments\environment.ts
  create src\favicon.ico
  create src\index.html
  create src\main.ts
  create src\polyfills.ts
  create src\styles.css
  create src\test.ts
  create src\tsconfig.json
  create .angular-cli.json
  create e2e\app.e2e-spec.ts
  create e2e\app.po.ts
  create e2e\tsconfig.json
  create .gitignore
  create karma.conf.js
  create package.json
  create protractor.conf.js
  create tslint.json
Directory is already under version control. Skipping initialization of git.
Installing packages for tooling via npm.
Installed packages for tooling via npm.
Project 'ClientApp' successfully created.

C:\Projects\YouFlyTest\YouFly.Web\YouFly.Web>
```

It should appear in your solution in visual studio:
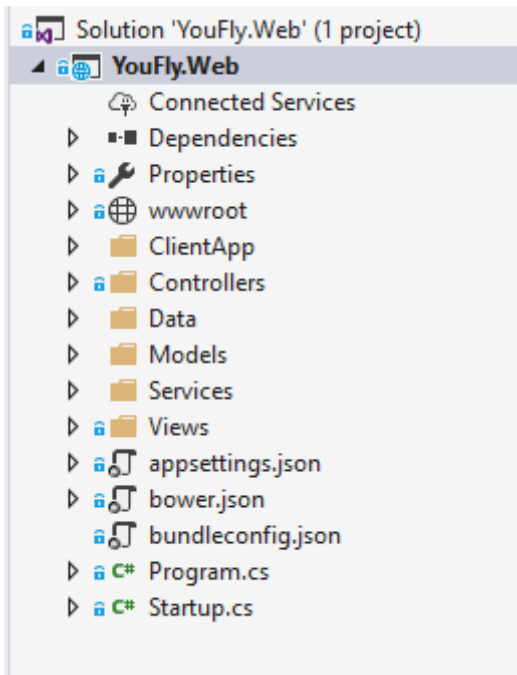
Right Click "ClientApp" then click "Exclude From Project" (this allows the project to build, angular CLI builds angular)

Alter the "ClientApp/.angular-cli.json" file's "apps.outDir" property to deploy to "../wwwroot/clientapp"

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "project": {
    "version": "1.0.0-beta.32.3",
    "name": "client-app"
  },
  "apps": [
    {
      "root": "src",
      "outDir": "../wwwroot/clientapp",
      "assets": [
        "assets",
        "favicon.ico"
```
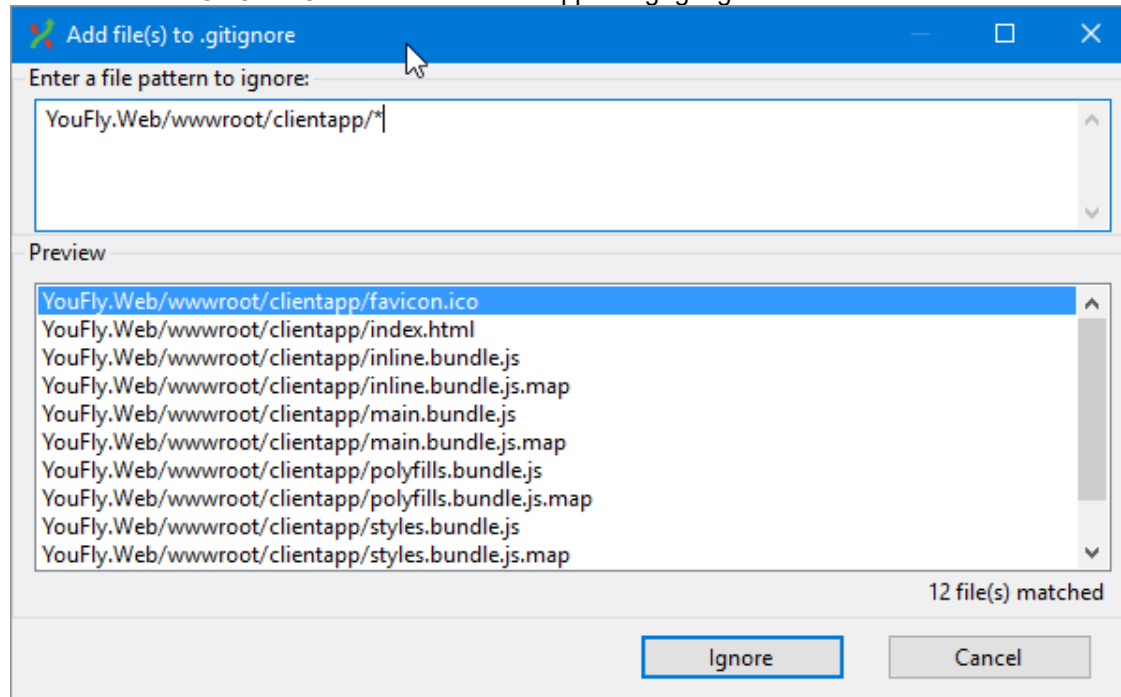
Now you can do "ng build --watch --base-href /clientapp/" and the angular 2 finished files will be in a folder and language that NET understands:
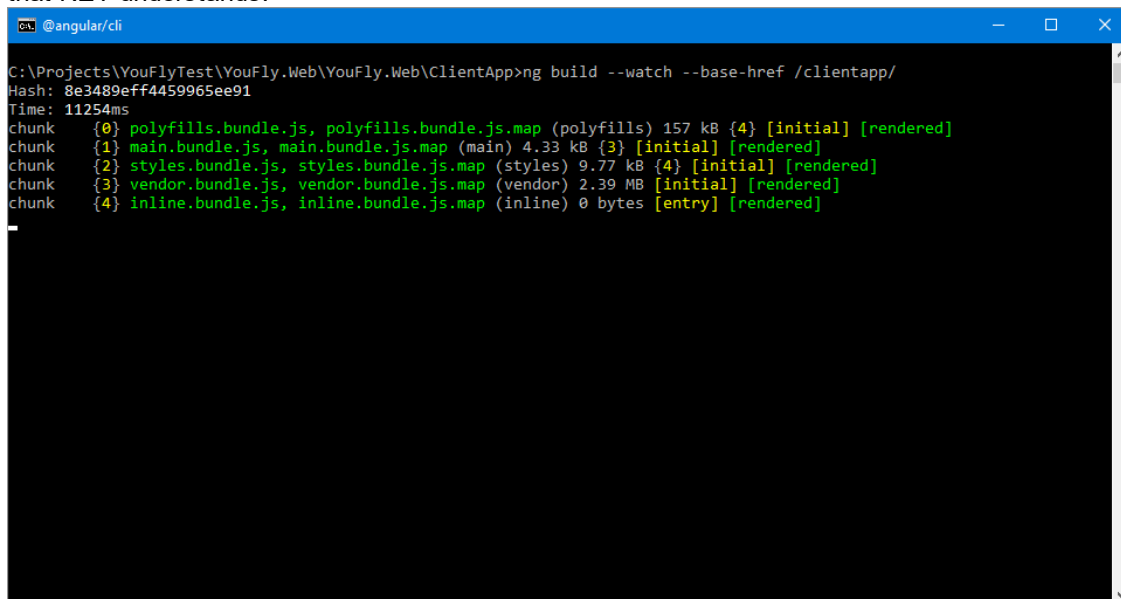
This PC  >  Local Disk (C:)  >  Projects  >  YouFlyTest  >  YouFly.Web  >  YouFly.Web  >  wwwroot  >

| Name | Date modified | Type | Size |
|------|---------------|------|------|
| clientapp | 3/15/2017 2:45 PM | File folder | |
| css | 3/15/2017 2:44 PM | File folder | |
| images | 3/15/2017 2:44 PM | File folder | |
| js | 3/15/2017 2:44 PM | File folder | |
| lib | 3/15/2017 2:44 PM | File folder | |
| favicon.ico | 3/15/2017 2:44 PM | Icon | 32 KB |

I'd recommend IGNORING the wwwroot client app using .git ignore

**Add file(s) to .gitignore**  —  □  ✕

Enter a file pattern to ignore:

YouFly.Web/wwwroot/clientapp/*|

Preview

```
YouFly.Web/wwwroot/clientapp/favicon.ico
YouFly.Web/wwwroot/clientapp/index.html
YouFly.Web/wwwroot/clientapp/inline.bundle.js
YouFly.Web/wwwroot/clientapp/inline.bundle.js.map
YouFly.Web/wwwroot/clientapp/main.bundle.js
YouFly.Web/wwwroot/clientapp/main.bundle.js.map
YouFly.Web/wwwroot/clientapp/polyfills.bundle.js
YouFly.Web/wwwroot/clientapp/polyfills.bundle.js.map
YouFly.Web/wwwroot/clientapp/styles.bundle.js
YouFly.Web/wwwroot/clientapp/styles.bundle.js.map
```

12 file(s) matched

Ignore          Cancel

(This will make sure you always build latest when you start to work)

Starting the visual studio "YouFly.Web" project:

Inline image 9
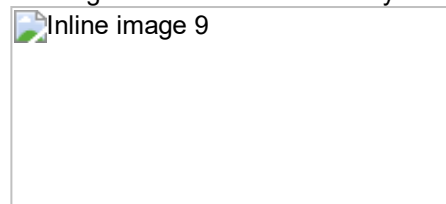
You'll see the default NET app:

YouFly.Web    Home    About    Contact                Register    Log in

There are powerful new features in Visual Studio for building modern web apps.   Learn More

### Application uses

- Sample pages using ASP.NET Core MVC
- Bower for managing client-side libraries
- Theming using Bootstrap

### How to

- Add a Controller and View
- Manage User Secrets using Secret Manager.
- Use logging to log a message.
- Add packages using NuGet.
- Add client packages using Bower.
- Target development, staging or production environment.

### Overview

- Conceptual overview of what is ASP.NET Core
- Fundamentals of ASP.NET Core such as Startup and middleware.
- Working with Data
- Security
- Client side development
- Develop on different platforms
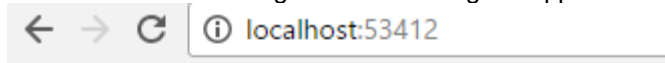- Read more on the documentation site

### Run & Deploy

- Run your app
- Run tools such as EF migrations and more
- Publish to Microsoft Azure Web Apps

Let's make the default page our angular app. Go to the home controller and replace "index" action with this:
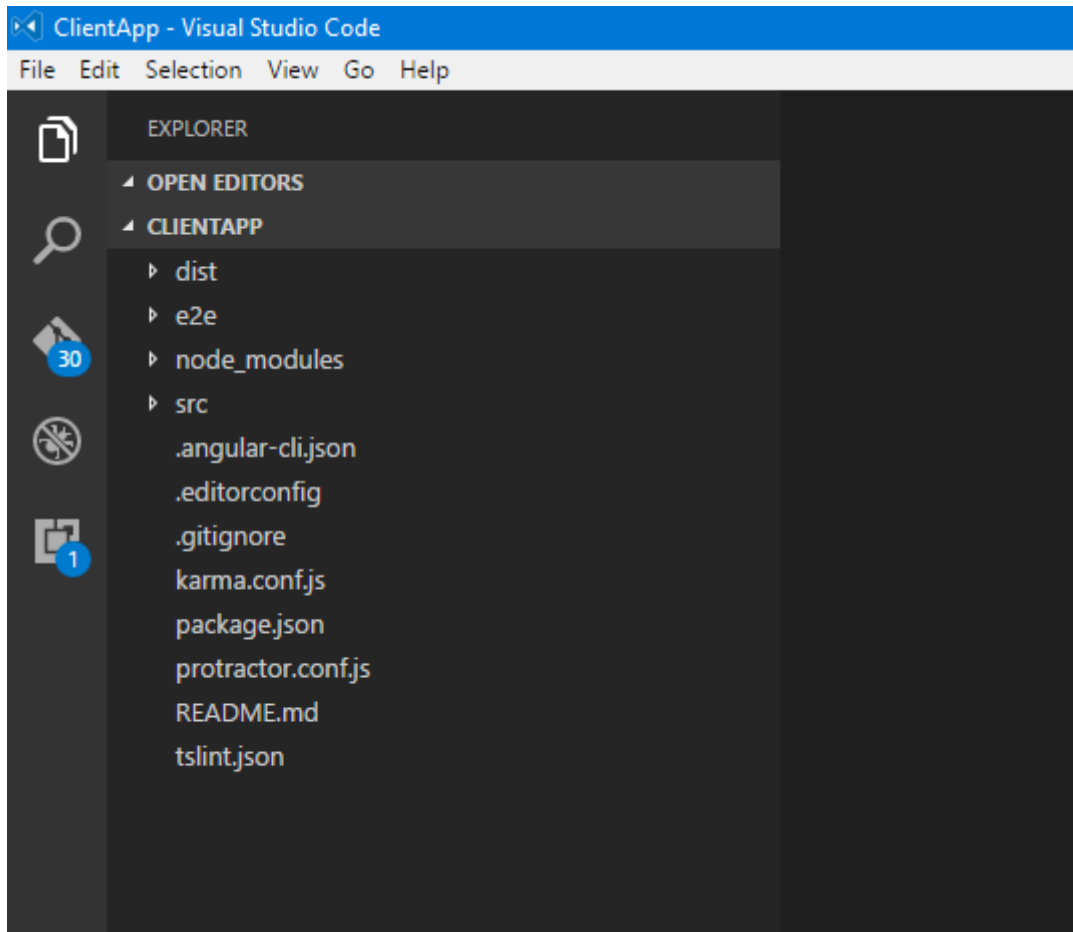
```
public IActionResult Index()
{
    return File("clientapp/index.html","text/html");
}
```

Build the VS solution again and then angular appears:

localhost:53412

# app works!

You can now use VS code to edit angular 2:

ClientApp - Visual Studio Code

File   Edit   Selection   View   Go   Help

EXPLORER

▲ OPEN EDITORS

▲ CLIENTAPP

  ▷ dist

  ▷ e2e

  ▷ node_modules

  ▷ src

  .angular-cli.json

  .editorconfig

  .gitignore

  karma.conf.js

  package.json

  protractor.conf.js

  README.md

  tslint.json

You can then edit here and see the changes with a MANUAL page refresh, e.g.:

Inline image 8

Inline image 9

← → C   ⓘ localhost:53412
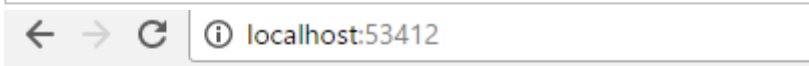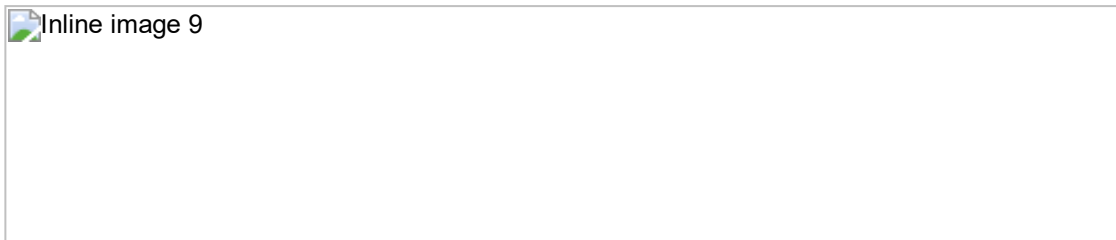
# app works! hello friends!

On Wed, Mar 15, 2017 at 1:49 PM, Andrew Ilgenfritz <andrew.ilgenfritz@selu.edu> wrote:

Yes it does.  How about this question. It's a question of development.  I've been doing some research and its a toss up right now.  The question is build angular 2 and add .net core on the back or build .net core then add angular2 on the front.

On Wed, Mar 15, 2017 at 1:34 PM, Matthew Vidacovich <mvid@envoc.com> wrote:
> VS Code is very good at working with angular 2
> Visual Studio is required to do xamarin
> Visual Studio is better at .NET Core MVC / Web Api things
> Visual Studio is not as good as VS Code at angular 2
>
> In short: Visual Studio is required, VS Code makes your JS development nicer but is not required
>
> Does that help?
>
> On Wed, Mar 15, 2017 at 12:59 PM, Andrew Ilgenfritz <andrew.ilgenfritz@selu.edu> wrote:
>> This is probably a dumb question but i'm going to ask it because i need the clarification.  for the web app portion of the final phase should we be configuring front end AND back end in VS Code?  I ask because I found options to do it all in VS Code or all in VS 2017 but not splitting it in both. I may be misunderstanding the whole thing.
>>
>>
>> --
>> **Matthew Vidacovich**
>> *Developer*

| Envoc | | PHONE: | (225) 910-8239 x106 |
| | | EMAIL: | mvid@envoc.com |
| Envoc | | | Fac Tw Lin Go RSS |

--
**Matthew Vidacovich**
*Developer*

| envōc<br>a better reality | PHONE: | (225) 910-8239 x106 |
| | EMAIL: | mvid@envoc.com |

--
**Matthew Vidacovich**
*Developer*

| envōc<br>a better reality | PHONE: | (225) 910-8239 x106 |
| | EMAIL: | mvid@envoc.com |

---

**Andrew Ilgenfritz** <andrew.ilgenfritz@selu.edu>                                    Wed, Mar 15, 2017 at 3:41 PM
To: Matthew Vidacovich <mvid@envoc.com>
Cc: 383 <383@envoc.com>

Thank you.  This is a great jumping off point.  Out of the 5 tutorials I found this is nothing like any of them.

[Quoted text hidden]

---

**Matthew Vidacovich** <mvid@envoc.com>                    Wed, Mar 15, 2017 at 5:15 PM
To: Andrew Ilgenfritz <andrew.ilgenfritz@selu.edu>
Cc: 383 <383@envoc.com>

Thank you for starting on this quickly!

And yeah, I had to scrap that together when you asked - if you find a way that works better for your team, go for it! That was just the first way I figured out when you asked.

[Quoted text hidden]

---

**Andrew Ilgenfritz** <andrew.ilgenfritz@selu.edu>                    Wed, Mar 15, 2017 at 6:20 PM
To: William Williamson <william.williamson-2@selu.edu>, Jeetendra Khadka <jeetendra.khadka@selu.edu>, Chad Bender <chad.bender@selu.edu>, aayush.dahal@selu.edu

Okay here is what vdoc sent me.  What I am going to do is set up a project locally and test some concepts. Specifically importing the csv file and searching it.I would suggest you do the same just to get used to the environment. It doesn't matter what you do with it because I'll push a fresh project to the repository probably early next week.  We will call this our research/test phase.  A subphase to the final phase if you will.

[Quoted text hidden]

---

**Andrew Ilgenfritz** <andrew.ilgenfritz@selu.edu>                    Fri, Mar 17, 2017 at 8:53 AM
To: Matthew Vidacovich <mvid@envoc.com>
Cc: 383 <383@envoc.com>

I hate asking little questions like this but I've spent the prescribed 2 hours hung up on it.  Excluding files in Visual studio 2017 was super easy. right-click, exclude done.  However, my group voted and we are going to be using 2015.  Heres the rub.  No right-click and exclude option.  I also tried to hide the folder in file explorer (someone on stack overflow said that would work).  I also tried adding buildOptions in the project.json with the "exclude" tag and listing my angular files.  I don't know 100% if it's not excluding files but when I build the project in visual studio it spits out errors because of files in my angular folder.  It's probably something easy but none of us can figure it out.

[Quoted text hidden]

---

**Matthew Vidacovich** <mvid@envoc.com>                    Fri, Mar 17, 2017 at 9:07 AM
To: Andrew Ilgenfritz <andrew.ilgenfritz@selu.edu>
Cc: 383 <383@envoc.com>

Honestly, I don't envy trying to use 2015 for this.

I don't think there is a way of removing it from the list in the project, but this *might* help:
https://github.com/aspnet/Announcements/issues/175
[Quoted text hidden]

---

**Andrew Ilgenfritz** <andrew.ilgenfritz@selu.edu>                    Fri, Mar 17, 2017 at 9:09 AM
To: Matthew Vidacovich <mvid@envoc.com>

lol that's the thread i found already to edit the project.json.  No big deal.  I'm gonna make an executive decision to use 2017

[Quoted text hidden]

---

**Matthew Vidacovich** <mvid@envoc.com>                    Fri, Mar 17, 2017 at 9:22 AM
To: Andrew Ilgenfritz <andrew.ilgenfritz@selu.edu>

Just so you understand - the excluding of those files was to prevent them from being compiled in VS (or rather, breaking the compile).
So, the compile : { exclude: [...] } thing would allow VS to build

[Quoted text hidden]