Using Angular in ASP.NET MVC 5 with Angular CLI and Visual Studio 2017

Angular is one of most in-demand web front-end frameworks developed by Google, it gets integrated with any Server side technology.
In this article, let's learn how to use Angular (Version 6) with ASP.NET MVC 5 using Angular-CLI.

I felt it was little tedious to get it working as so many technologies are involved. This article describes minimal steps to get started.

Software pre-requites

- Microsoft Visual Studio 2017 (Community Edition)
- Install Latest NodeJs
- TypeScript 2.6 minimum.

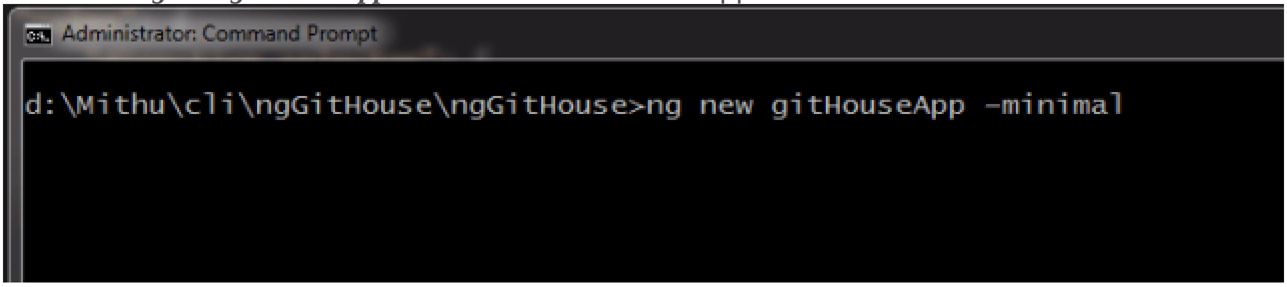
Installing Angular CLI

Angular CLI is a tool for developing an Angular-based (web, PWA) application, everything is out of the box like generating components, services, pipes, unit tests etc.
For installing CLI, its must-have NodeJS installed previously. Use this command to install CLI

```
npm install -g @angular/cli
```

Create ASP.NET MVC 5 & Angular app together

Create an ASP.NET MVC 5 application. Named it as *ngGitHouse*. Nothing fancy in this but its first step.
Once the CLI is installed, we create a brand new Angular application by running this command ***ng new gitHouseApp –minimal*** inside MVC 5 application folder structure.



```
d:\Mithu\cli\ngGitHouse\ngGitHouse>ng new gitHouseApp -minimal
```

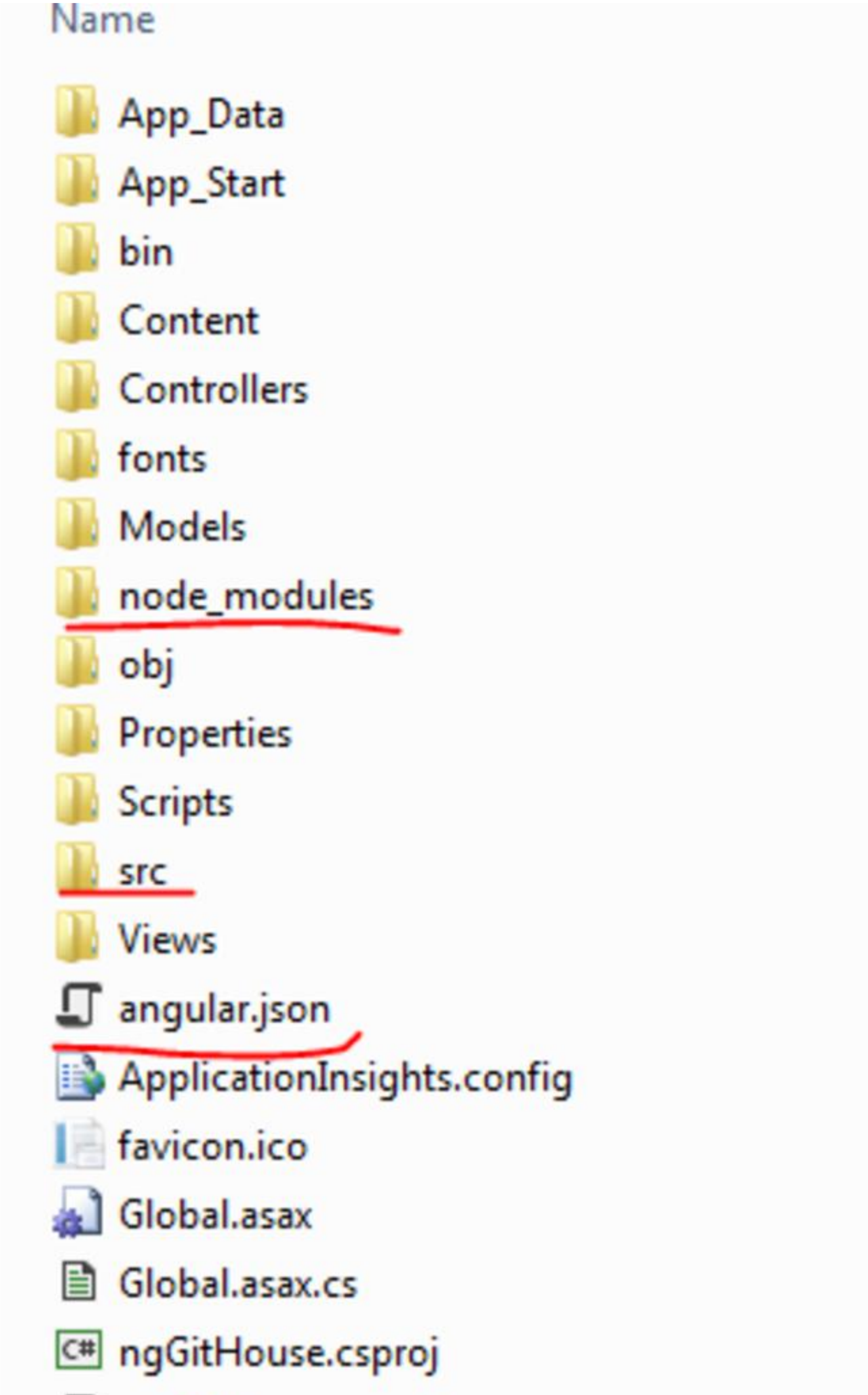
Create Angular app inside MVC 5

It would take few minutes to get all node modules downloaded. You can see in the folder structure that *githouseapp* folder is created in MVC 5 application folder structure.
To ensure that the *githouseapp* is set up properly; navigate to folder path in command prompt and run the following command **ng build**. If this succeeds then your good to go.

Moving essentials files & folder to root

Our intention is to use Angular framework inside MVC 5 views, to make it easy for understanding and maintenance lets move some essentials files and folder to root of MVC 5 application. The files & folder to be moved are

- **Src** folder - This is actual source folder of the Angular application, entire project structure is present in this folder.
- **package.json -** file containing the list of NPM packages needed to develop client application
- **angular.json -** file containing Configuration settings for the Angular application. This file is essential for Angular-CLI to work seamlessly.
- **tsconfig.json -** configuration file must for all TypeScript files to transpile to JavaScript.
- **node_modules -** folder containing all downloaded node modules. This folder is always heavy.



Name

App_Data
App_Start
bin
Content
Controllers
fonts
Models
node_modules
obj
Properties
Scripts
src
Views
angular.json
ApplicationInsights.config
favicon.ico
Global.asax
Global.asax.cs
ngGitHouse.csproj

ASP.NET MVC 5 & Angular files together

Update configuration settings

We have altered how AngularCLI generates the folder structure because we indent to use it in ASP.NET MVC 5 application.

For CLI to work well, we have to update settings in the below configuration file **tsconfig.json** - This file is used by TypeScript compiler to transpile to JavaScript. It's a mandatory file wherever TypeScript is used.

The ***include*** config entry tells us to compile TypeScript from **src** folder only instead of entire project structure. If you plan to use TypeScript in another folder, do add in this section.

The ***outDir*** entry in *compilerOptions* provides a folder for placing all transpiled files with source maps. Source Maps helps us to debug the TypeScript (Angular in this case) code in the browser just like JavaScript code.

```
{
    "compileOnSave": false,
    "include": [
      "./src"
    ],
    "compilerOptions": {
      "baseUrl": "./",
      "outDir": "./scripts/out-tsc",
      "sourceMap": true,
      "declaration": false,
      "moduleResolution": "node",
      "emitDecoratorMetadata": true,
      "experimentalDecorators": true,
      "target": "es5",
      "typeRoots": [
        "node_modules/@types"
      ],
      "lib": [
        "es2017",
        "dom"
      ]
    }
}
```

angular.json - This is the heart of AngularCLI, it contains all options necessary to play around Angular artifacts like generating components, pipes, service provider, class, directives etc.

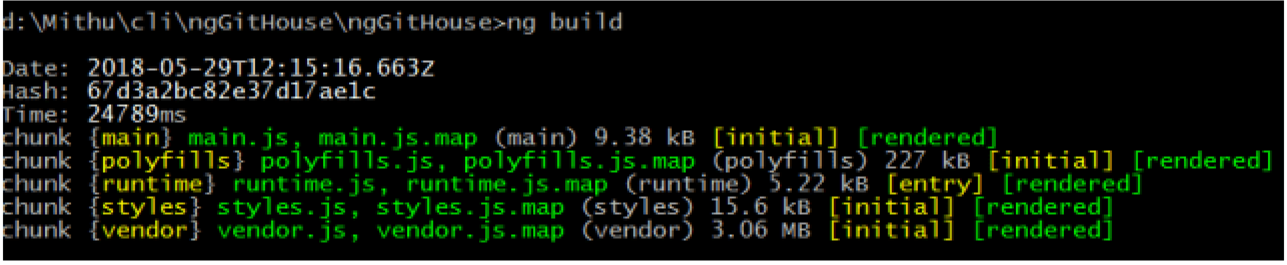
The *sourceRoot* now points to *src* folder, the ***outputPath*** is now pointing to **./Scripts/libs** folder as part of MVC 5 project. The output files of *ng build command* will be copied here.

*Recommend removing the **githouseapp** created by CLI project.*

Building the application

As we moved folder location, configuration files got updated, its best to run the command ***ng build*** in project root folder from command prompt.

If done successfully, you would see a similar image as below. *Don't forget to include **scripts/libs**folder in Solution Explorer.*
***ng build –watch** will run the build when file changes*



```
d:\Mithu\cli\ngGitHouse\ngGitHouse>ng build

Date: 2018-05-29T12:15:16.663Z
Hash: 67d3a2bc82e37d17ae1c
Time: 24789ms
chunk {main} main.js, main.js.map (main) 9.38 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 227 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 5.22 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 15.6 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 3.06 MB [initial] [rendered]
```

ng build running successfully

Loading Angular in MVC 5 views

Now that everything is building properly, let's load the Angular app in ASP.NET MVC 5 views. Will be using ShowAngularPage.cshtml file of HomeController's ShowAngualrPage action method generated while scaffolding MVC application.

```
using System.Web.Mvc;

namespace ngGitHouse.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            ViewBag.Message = "Your application description page.";

            return View();
        }

        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";

            return View();
        }

        public ActionResult ShowAngularPage()
        {
            return View();
        }
    }
}
```

Removed the existing code to include our code to load the component as below in ShowAngularPage.cshtml.
The **app-root** is an Angular Component generated by default using CLI

```
@section
Scripts
{
        <script  type="text/javascript"  src="~/Scripts/libs/runtime.js"></script>
        <script  type="text/javascript"  src="~/Scripts/libs/polyfills.js"></script>
        <script  type="text/javascript"  src="~/Scripts/libs/styles.js"></script>
        <script  type="text/javascript"  src="~/Scripts/libs/vendor.js"></script>
        <script  type="text/javascript"  src="~/Scripts/libs/main.js"></script>
    }
    <app-root></app-root>
```

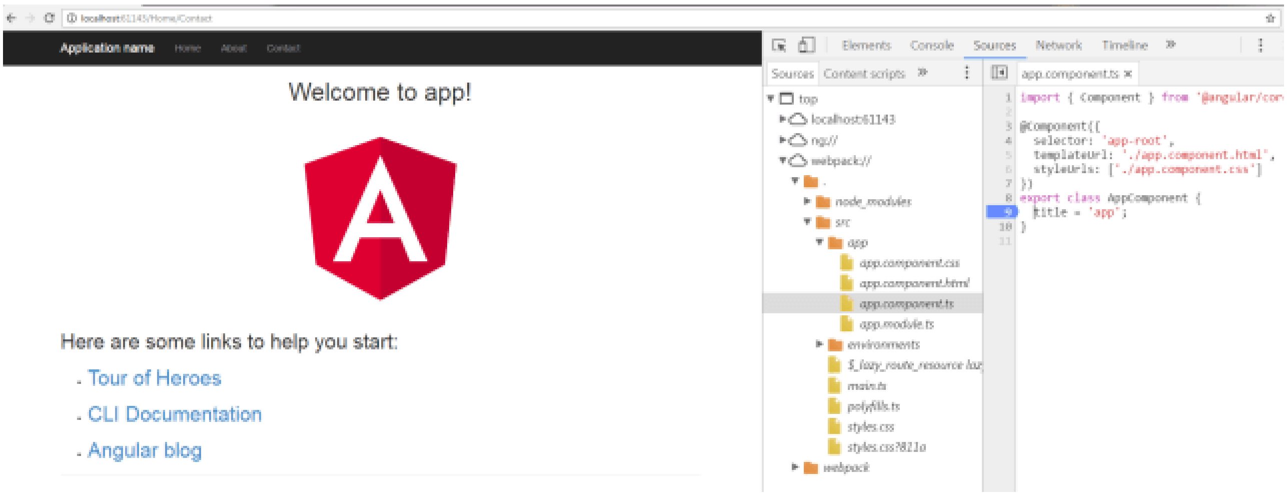
The *Scripts* section includes the link to files created in **libs** folder run from above step. **The JS files referencing order is important here.** Add the follwing code snippet in _Layout.cshtml page for navigation link to Angualr embedded page.

```
@Html.ActionLink("Angular MVC Git House WebApp", "ShowAngularPage", "Home", new { area
= "" }, new { @class = "navbar-brand" })
```

Run the application, click on the '`Angular MVC Git House WebApp`' link on the navbar to load the Angular.

Debugging the app

We successfully ran the Angular code in ASP.NET MVC 5, debugging the code in the browser (chrome) involves press F12, select *Sources* tab.
Check out below image for file location while running application.
Do run the command ***ng build –watch*** to compile Angular code automatically



```
import { Component } from '@angular/cor
 
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'app';
}
```

Running & Debugging Angular Code in ASP.NET MVC 5