



mithunvp.com

ASP.NET Core

FOLLOW:



ANGULAR / ASP.NET MVC 5

PREVIOUS STORY



Building fully Asynchronous
ASP.NET Core Web API

Search

RECENT POSTS

🕒 Using Angular 9 in ASP.NET MVC 5 with
Angular CLI and Visual Studio 2017/ 2019
May 29, 2018

🕒 Building fully Asynchronous ASP.NET
Core Web API
August 23, 2017

🕒 Working with VueJS 2 and ASP.NET
MVC 5 in Visual Studio
August 16, 2017

🕒 Using TypeScript with ASP.NET MVC 5
instead of JavaScript
August 7, 2017

🕒 Building ASP.NET Core MVC as SPA
using Angular 2 and TypeScript
October 27, 2016

Using Angular 9 in ASP.NET MVC 5 with Angular CLI and Visual Studio 2017/ 2019

BY [MITHUNVP](#) · PUBLISHED MAY 29, 2018 ·
UPDATED AUGUST 9, 2020

Angular is one of the most in-demand web front-end frameworks developed by Google. It gets integrated with

34



0



0



0



0



CATEGORIES

- Angular
- Angular 2
- ASP.NET Core
- ASP.NET Core 1.0
- ASP.NET MVC 5
- Async
- Dot Net
- EF Core 2
- Install & Configure
- LINQ
- TypeScript
- Visual Studio Code
- WCF

TAGS

Angular **Angular2** asp.net core
Asp.NET Core 1.0
 ASP.NET Core 2 ASP.NET MVC5
 ASP.NET MVC 5 Async Await **C#** EF
 Core 2 middleware MVC 5 Sql Server
TypeScript Visual Studio Visual
 Studio Code VueJS 2 Windows 10

this article, let
 use Angular (C
 ASP.NET MVC 5
 CLI.

ASP.NET CORE
**Building fully
 Asynchronous
 ASP.NET Core Web
 API**

I had written c
 Angular4 in [ASP.NET MVC 5](#), I
 felt it was a little tedious to get
 it working as so many
 technologies are involved. This
 article describes minimal
 steps to get started.

“ *Upgraded to use Angular
 9 with ASP.NET MVC 5 on
 branch **mvc5angular9***

Software pre-requisites

- Microsoft Visual Studio
2017/ 2019 (any edition)
- Install Latest [NodeJs](#)
- TypeScript 3.8

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

34

0 0 0 0



“ npm install @angular

ASP.NET CORE

Building fully
Asynchronous
ASP.NET Core Web
API

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.



Create an Angular app inside MVC 5

It would take a 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;

34

0 0 0



command pr
following com
If this succee
good to go.

ASP.NET CORE
Building fully
Asynchronous
ASP.NET Core Web
API

Moving essentials files & folder to root

Our intention is to use the 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 the actual source folder of the Angular application, the 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

34

0 0 0 0



all TypeScript
transpile to
■ **node_modules**
containing
node modules
always heavy.

ASP.NET CORE
Building fully
Asynchronous
ASP.NET Core Web
API

“ Do NOT forget to include
the above files & folder in
Solution Explorer except
node_modules

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
- ngGitHouse.csproj.user
- package.json
- package-lock.json
- packages.config
- tsconfig.json
- Web.config
- Web.Debug.config
- Web.Release.config

ASP.NET MVC 5 & Angular files

34

0 0 0



ASP.NET CORE

Building fully
Asynchronous
ASP.NET Core Web
API

Update config 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 the **src** folder only instead of the 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

34  0 0 0

```
{
  "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"
    ]
  }
}
```

ASP.NET CORE

Building fully
Asynchronous
ASP.NET Core Web
API

angular.json – This is the heart of AngularCLI, it contains all options necessary to play around Angular artifacts like generating components, pipes, service providers, class, directives, etc. The *sourceRoot* now points to the *src* folder, the ***outputPath*** is now pointing to ***./Scripts/libs*** folder as part of the MVC 5 project. The output files of the *ng build*

```
"architect": {
  "build": {
    "builder": "@angular-devkit/build-angular",
    "options": {
      "outputPath": "dist",
      "index": "src/index.html"
    }
  }
}
```

ASP.NET CORE

Building fully
Asynchronous
ASP.NET Core Web
API

OutputPath updated

I recommend

*the **githubseapp** created by
the CLI project.*

Building the application

As we moved folder location, configuration files got updated, its best to run the command **ng build** in the project root folder from the 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: 2478ms
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

34

0

0

0

0

<

Now that everything is set up properly, let's build our Angular9 application. I will be editing *Contact.cshtml* while scaffolding MVC.

ASP.NET CORE
Building fully
Asynchronous
ASP.NET Core Web
API

application. I removed the existing code to include our code to load the component as below. 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

34

0 0 0 0



Run the applic
the Contact li
to load the Ar

[Source code](#)
[Github](#)  [account](#)

to play around with it

ASP.NET CORE

Building fully
Asynchronous
ASP.NET Core Web
API

Debugging the app

We successfully ran the Angular9 code in ASP.NET MVC 5, debugging the code in the browser (chrome) involves press F12, select *Sources* tab. Check out the below image for file location while running application. Do run the command ***ng build --watch*** to compile Angular9 code automatically

*Running & Debugging Angular Code
in ASP.NET MVC 5*

34 

 0



 0

 0



Tags: Angular



Working
with
VueJS 2
and
ASP.NET
MVC 5
in Visual
Studio

AUGUST
16, 2017

ASP.NET CORE

Building fully
Asynchronous
ASP.NET Core Web
API

Sponsored

Getting this Treas

Hero Wars

Prendi un fazzolet
all'età di 57 anni

LawyersFavorite

ASP.NET CORE

Building fully
Asynchronous
ASP.NET Core Web
API

Terribili foto del Titanic trovate su una vec
fotografica..

Novelodge

Cosa sarebbe successo se aveste investit
anno fa?

eToro

Il figlio di Cristina Parodi è probabilmente
pianeta

Wordsa

5 siti d'incontri grazie ai quali avrai un suc

Top 5 Dating IT

I 25 eserciti più potenti al mondo

Happy Landings

ALSO ON WWW.MITHUNVP.COM

2 years ago • 11 comments

Deploying
Angular with
ASP.NET MVC 5

2 years ago

Angular
Element
to use

Comments Community Privacy

Recommend Tweet Share

Start the discussion...

34

f

t

i

r



ASP.NET CORE

Building fully
Asynchronous
ASP.NET Core Web
API

OR SIGN UP WITH DISC

Name

Be the first to comme

ASP.NET Core



mithunvp.com © 2020. All Rights
Reserved.

Powered by  - Designed with the Hueman
theme

