Make your Angular App SEO-Friendly (Angular 4 + Universal)

BY GARY SIMON - MAY 18, 2017

The Github Repo for this Tutorial: Visit the repo

By default, Angular apps are not search engine friendly. If you view the page source in a browser of a regular Angular app, it will only show what's inside the regular index.html -- *the infamous* ***loading..*** content. To fix this, we need the app to be rendered on a server and then displayed to the client. This way, your content for each page (each component) is spit out for the browser and that lovely Googlebot to see.

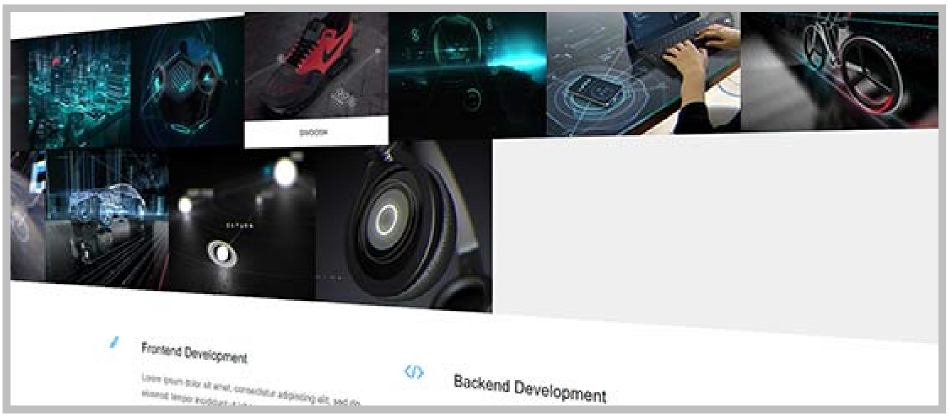
But to a beginner-intermediate Angular developer, the topic of Angular SEO is usually a confusing one. Quick Google searches reveal outdated information and tutorials that no longer work.

Fortunately, I'm here to do my absolute best to make everything easy to understand and walk you through the setup process through this written tutorial, and a video tutorial that you can watch below (some people prefer watching instead of reading!).

So, let's get started.

Create a MEAN App Called CodePost - Full Stack

Learn Angular 4 from Scratch

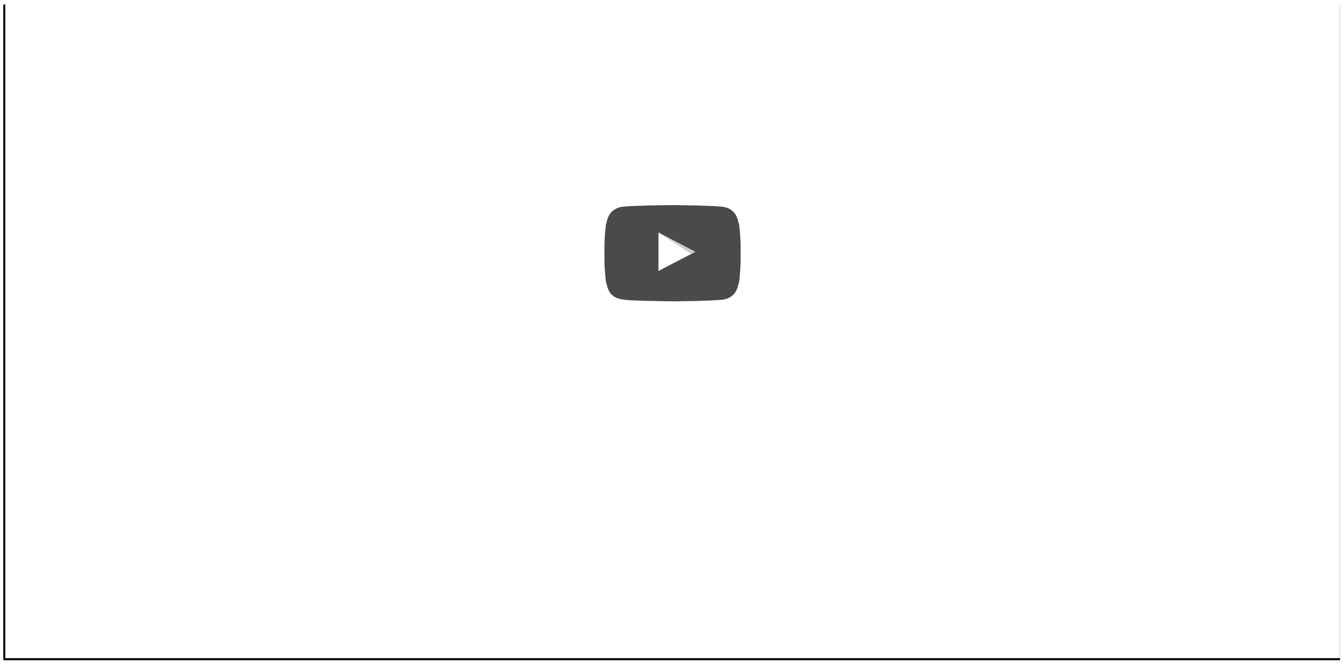


Create a Personal Portfolio using Angular 2 & Behance

If you prefer watching a video..

Be sure to Subscribe to the Official Coursetro Channel on Youtube for more awesome videos.

Angular 4 SEO Tutorial (Universal + Angular CLI)



Get the Project Up and Running

As usual, we will start the project with the Angular CLI. Check out our **Free Angular Course** if you're new to the CLI.

```
> ng new ang4-seo --routing
> cd ang4-seo
```

The **--routing** flag will generate a quick routes file and add it to our app module.

Once we're in the new project folder, we'll use npm to install the platform-server, which is needed for server side rendering and generating HTML pages. We also **must** install the animations package (new to Angular v4), otherwise your app will result in an error:

```
> npm install --save @angular/platform-server @angular/animations
```

Next, we'll head into **/src/app/app.module.ts** and make a slight adjustment:

```
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule.withServerTransition({appId: 'ang4-seo-pre'}),
    FormsModule,
    HttpModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
```

The only line that is changed is the BrowserModule. Here, we've added a .withServerTransition which takes in an appId that is shared between the client and the server. .withServerTransition allows Universal to replace the HTML it generated on its own.

Save this file.

Create a Server App Module

Create a new file called /src/app/app.server.module.ts and paste in the following code:

```
import { NgModule } from '@angular/core';
import { ServerModule } from '@angular/platform-server';
import { AppModule } from './app.module';
import { AppComponent } from './app.component';

@NgModule({
imports: [
    ServerModule,
    AppModule
],
bootstrap: [AppComponent]
})
export class AppServerModule { }
```

This is used to bootstrap the app from the server. If you compare it against /src/app/app.module.ts, they're very similar. The main takeaway here is that this module is for the server.

Creating the Express Server

Naturally, we need a server to render the app before pushing it to the client. So, create a new file called /src/server.ts and paste the following:

```
import 'reflect-metadata';
import 'zone.js/dist/zone-node';
import { platformServer, renderModuleFactory } from '@angular/platform-server'
import { enableProdMode } from '@angular/core'
import { AppServerModuleNgFactory } from '../dist/ngfactory/src/app/app.server.n
import * as express from 'express';
import { readFileSync } from 'fs';
import { join } from 'path';

const PORT = 4000;

enableProdMode();

const app = express();

let template = readFileSync(join(__dirname, '..', 'dist', 'index.html')).toStrin

app.engine('html', (_, options, callback) => {
  const opts = { document: template, url: options.req.url };

  renderModuleFactory(AppServerModuleNgFactory, opts)
    .then(html => callback(null, html));
});

app.set('view engine', 'html');
app.set('views', 'src')

app.get('*.*', express.static(join(__dirname, '..', 'dist')));

app.get('*', (req, res) => {
  res.render('index', { req });
});

app.listen(PORT, () => {
  console.log(`listening on http://localhost:${PORT}!`);
});
```

I won't go into detail about everything that's happening here, as much of it is specific to Express and extends outside of the scope of this tutorial.

But, the heart of what's specific to server-side rendering (Universal) here is the **renderModuleFactory** method.

It takes in **AppServerModuleNgFactory** that the server generates after building the project, which is a compiled Angular app:

```
import { AppServerModuleNgFactory } from '../dist/ngfactory/src/app/app.server.r
```

Note that it's located within the standard **/dist** folder that's generated after running the **ng build** command. Then it serves it as HTML.

TypeScript Config Adjustments

Open **/src/tsconfig.app.json** and add the **server.ts** file we created earlier as an exclude:

```
{
  "extends": "../tsconfig.json",
  "compilerOptions": {
    "outDir": "../out-tsc/app",
    "module": "es2015",
    "baseUrl": "",
    "types": []
  },
  "exclude": [
    "server.ts",     // Right here
    "test.ts",
    "**/*.spec.ts"
  ]
}
```

Then, open **/tysconfig.json** and add **angularCompilerOptions**:

```
{
  "compileOnSave": false,
  "compilerOptions": {
    "outDir": "./dist/out-tsc",
    "baseUrl": "src",
    "sourceMap": true,
    "declaration": false,
    "moduleResolution": "node",
    "emitDecoratorMetadata": true,
    "experimentalDecorators": true,
    "target": "es5",
    "typeRoots": [
      "node_modules/@types"
    ],
    "lib": [
      "es2016",
      "dom"
    ]
  },
  "angularCompilerOptions": {                             // Here
      "genDir": "./dist/ngfactory",
      "entryModule": "./src/app/app.module#AppModule"
  }
}
```

angularCompilerOptions allows us to specify a genDir property, which is where the generated ngfactory files for the components and modules will go, and an entryModule which accepts the path of our main bootstrapped module. The #AppModule at the end of the path is the name of the exported class.

Adjust the Run Scripts

Inside /package.json we'll adjust the scripts property:

```
{
  //
  // Other properties removed for brevity
  //

  "scripts": {
    "prestart": "ng build --prod && ngc",
    "start": "ts-node src/server.ts"
  },

  //
  // Other properties removed for brevity
  //

}
```

We add a pre in front of the start command to first run the ng build --prod && ngc command. After that, it launches the server based on our server.ts config.

Now, you can execute the following command in the console to run the app!

```
  > npm run start
```

If all goes smooth, the output will look something like this:

```
  > ang4-seo-pre@0.0.0 prestart C:\Users\gary\code\ang4-seo-pre
  > ng build --prod && ngc

  Hash: 831974a6aced5d532e0d
  Time: 6395ms
  chunk    {0} polyfills.6ce0bf58446ef5c2585b.bundle.js (polyfills) 157 kB {4} [ir
  chunk    {1} main.94206ad8fdce0f306470.bundle.js (main) 21.3 kB {3} [initial] [r
  chunk    {2} styles.d41d8cd98f00b204e980.bundle.css (styles) 69 bytes {4} [initi
  chunk    {3} vendor.20c071d0dcfbd58f1c48.bundle.js (vendor) 1.1 MB [initial] [re
  chunk    {4} inline.6729d86a888b9e64ae89.bundle.js (inline) 0 bytes [entry] [rer

  > ang4-seo-pre@0.0.0 start C:\Users\gary\code\ang4-seo-pre
  > ts-node src/server.ts

  listening on http://localhost:4000!
```

If you visit http://localhost:4000 in a browser, you can view the page source and now the source actually shows app works! when it would typically show the default Loading.. text defined in index.html.

Let's give it a better test and generate a couple components with routes. We'll even add custom title and meta tags for each page.

Generate Components

Using the CLI to generate components is easy, but when you run the standard command, such as ng g c home, you might run into an error:

```
> ng g c home
installing component
Error locating module for declaration
        SilentError: Multiple module files found:
```

To avoid this, we can specify the module by adding a flag. So, run the following commands to generate 2 components:

```
> ng g c home --module=app.module.ts
> ng g c about --module=app.module.ts
```

Setup the Routes

Let's hop into the /src/app/app-routing.module.ts file that the CLI generated with the --routing flag:

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { AboutComponent } from './about/about.component';
import { HomeComponent } from './home/home.component';

const routes: Routes = [
  {
    path: '',
    component: HomeComponent
  },
  {
    path: 'about',
    component: AboutComponent
  }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Here, we're importing our About and Home components that we just generated, and we're adding them each to **Routes**.

App Component Template

Let's visit the **/src/app.component.html** file and change it to this:

```
<ul>
  <li><a routerLink="/">Home</a></li>
  <li><a routerLink="about">About</a></li>
</ul>

<router-outlet></router-outlet>
```

We're creating a navigation with a **router-outlet** for our home and about components.

Adding Title and Meta Tags to our Components

A cornerstone of on-page SEO factors are unique titles, meta descriptions and meta keywords. Fortunately, the process of defining these elements is straightforward for each component.

In **/src/app/home/home.component.ts** import up top:

```
import { Meta, Title } from "@angular/platform-browser";
```

Then, in the constructor:

```
export class HomeComponent implements OnInit {

  constructor(meta: Meta, title: Title) {

    title.setTitle('My Spiffy Home Page');

    meta.addTags([
      { name: 'author',   content: 'Coursetro.com'},
      { name: 'keywords', content: 'angular seo, angular 4 universal, etc'},
      { name: 'description', content: 'This is my Angular SEO-based App, enjoy i
    ]);

  }

  ngOnInit() {
  }

}
```

We're using dependency injection to create an instance of both **Meta** and **Title**, then we're using both to set the titles and various meta tags that we wish to include for this component.

Save it.

About Component

Now, we'll repeat the same exact process above for the **/src/app/about/about.component.ts** file. Just be sure to change the title and meta tags to something unique so that we can see them change momentarily.

Let's also change the template slightly in **/src/app/about/about.component.html**:

```
<p>And this is my lovely description that I want the Google to see!</p>
```

Run it!

Visit the console, ctrl-c out if the server is still running and then run:

```
> npm run start
```

With any luck, it will run without error and you can view the page source of the first page and notice the title and 3 meta keywords are there as specified within the home component.

Then, click on the about link and check out the page source again. Wala!

Hopefully, this tutorial helped you out quite a bit and put you on the way to SEO success.

This tutorial is largely based on the example located here, however, it lacked any explanation. Kudos to Éverton Roberto Auler though!

Skip the manual setup stuff

The project that I just created can be cloned at Github. Just visit this link and you can be up n' running quickly!

Share this post

Chat with us

Say something about this awesome post!

34 Comments **Coursetro** **Armiie Armza**

♡ Recommend 5 Share Sort by Best

Join the discussion…

Thibault • 2 months ago

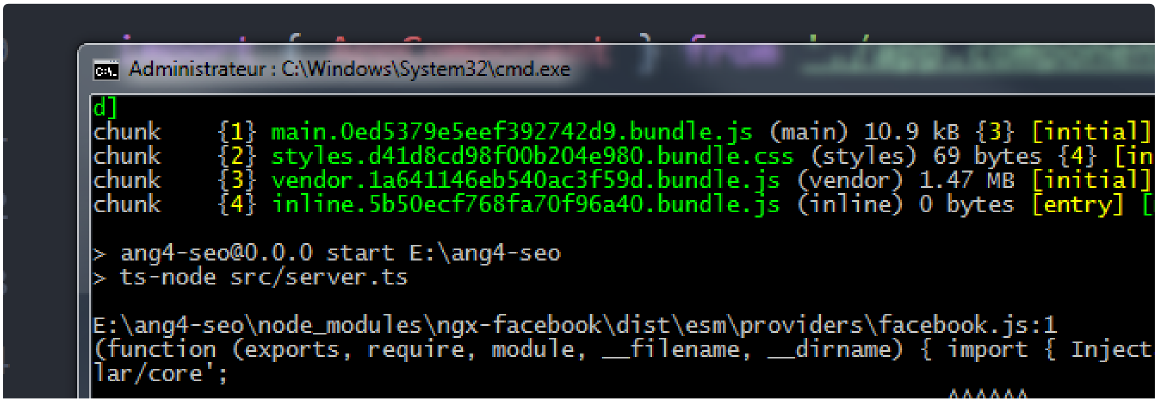
trying to implement that into my project. But when running ts-node src/server.ts, i'm getting this error :
"SyntaxError: Unexpected token import" in app.module.js.

Any idea ?

9 ^ | ˅ • Reply • Share ›

SAMSOL ➦ Thibault • a month ago

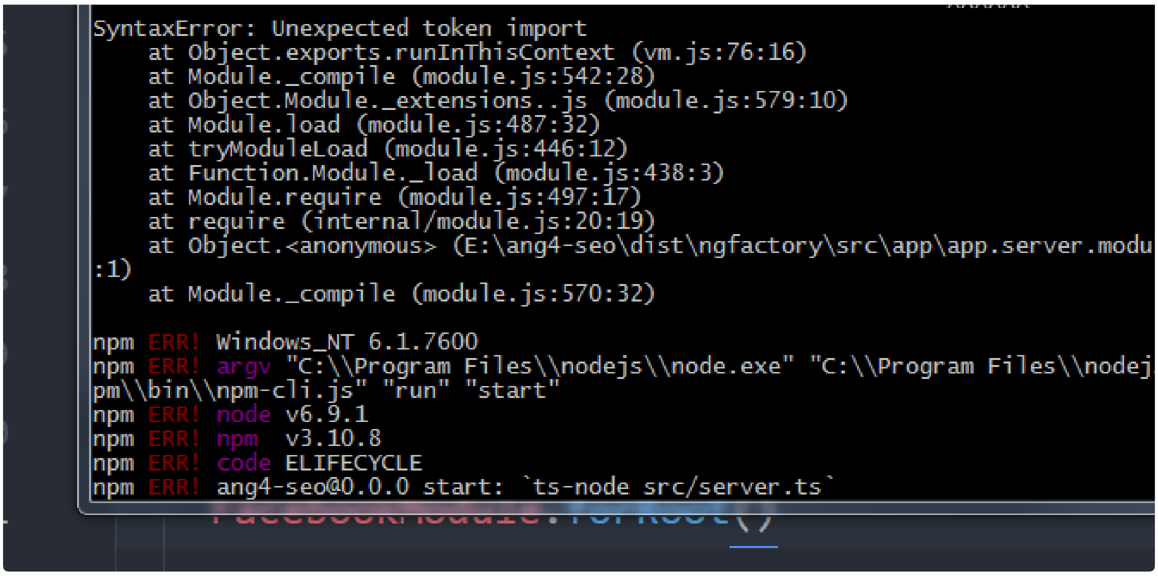
also me !!!
please there is answer



```
d]
chunk    {1} main.0ed5379e5eef392742d9.bundle.js (main) 10.9 kB {3} [initial]
chunk    {2} styles.d41d8cd98f00b204e980.bundle.css (styles) 69 bytes {4} [in
chunk    {3} vendor.1a641146eb540ac3f59d.bundle.js (vendor) 1.47 MB [initial]
chunk    {4} inline.5b50ecf768fa70f96a40.bundle.js (inline) 0 bytes [entry] [

> ang4-seo@0.0.0 start E:\ang4-seo
> ts-node src/server.ts

E:\ang4-seo\node_modules\ngx-facebook\dist\esm\providers\facebook.js:1
(function (exports, require, module, __filename, __dirname) { import { Inject
lar/core';
```



```
SyntaxError: Unexpected token import
    at Object.exports.runInThisContext (vm.js:76:16)
    at Module._compile (module.js:542:28)
    at Object.Module._extensions..js (module.js:579:10)
    at Module.load (module.js:487:32)
    at tryModuleLoad (module.js:446:12)
    at Function.Module._load (module.js:438:3)
    at Module.require (module.js:497:17)
    at require (internal/module.js:20:19)
    at Object.<anonymous> (E:\ang4-seo\dist\ngfactory\src\app\app.server.modu
:1)
    at Module._compile (module.js:570:32)

npm ERR! Windows_NT 6.1.7600
npm ERR! argv "C:\\Program Files\\nodejs\\node.exe" "C:\\Program Files\\nodej
pm\\bin\\npm-cli.js" "run" "start"
npm ERR! node v6.9.1
npm ERR! npm  v3.10.8
npm ERR! code ELIFECYCLE
npm ERR! ang4-seo@0.0.0 start: `ts-node src/server.ts`
```

^ | ˅ • Reply • Share ›

Ehsan Ershadi ➦ SAMSOL • 19 days ago

instead of 'npm run start' use :
> ng build --prod
> .\node_modules\.bin\ngc -p tsconfig.json
> ts-node src/server.ts

or
"prestart": "ng build --prod && call .\\node_modules\\.bin\\ngc -p tsconfig.json",

1 ^ | ˅ • Reply • Share ›

Noman Hasan • 2 months ago

Thank you very much for the awesome tutorial. But How do you import any third party library like angular material or ngx-bootstrap? When I try to use these libraries "Syntax Error: unexpected token ... " Error appears. Please share if you have any solution.

6 ^ | ˅ • Reply • Share ›

jagdish • 2 months ago

Thanks, Nice very helpful to me but I have question regrading how to deploy angular server site rending on aws ec2? can i install the angular cli on aws ec2? please help me which step i ill will follow for my app will wrok on aws ec2 to work.

2 ^ | ˅ • Reply • Share ›

Abava • 2 months ago

Is there a way to automatically refresh, or should we `npm start` every time we change something?

2 ^ | ˅ • Reply • Share ›

Devon • 2 months ago

Nice post, thanks. How would you go about deploying this to a cloud server for production?

1 ^ | ˅ • Reply • Share ›

Diego Martín ➦ Devon • a month ago

I'd also like to know how to deploy this to production's nodeJs. I have tried to release the angular universal app in Azure and documented it here but it didn't work. https://stackoverflow.com/q...

It seems it requires the server.ts to be already built and executed with node command directly. So it'd be great to get some tips or some links we could read on what are the steps. Cheers

⌃ | ⌄ • Reply • Share ›

Michalis Tzikas • 2 months ago

Hello, I run your instructions and it works like a charm. But... in my local enviroment. What I shoud do for my production. I simply upload the contents of my dist folder to the server. I guess that I have to run some node server now. Right? How??

1 ⌃ | ⌄ • Reply • Share ›

iwish Sedo ➦ Michalis Tzikas • a month ago

I'm facing the same issue. It works for me but only on local. Did you find a way for making it work in production?

⌃ | ⌄ • Reply • Share ›

Diego Martín ➦ Michalis Tzikas • a month ago

If you find out please let me know :) I've been struggling with it and I opened without success so far a question at stackoverflow: https://stackoverflow.com/q...

⌃ | ⌄ • Reply • Share ›

Dmytro Garastovych • 2 months ago

Thank's for the article, it's really a nice read. Unfortunately it doesn't work with scss, giving an error:
Error: Compilation failed. Resource file not found: ${PROJECT_PATH}/src/_colors
Path is correct

1 ⌃ | ⌄ • Reply • Share ›

Janusz ➦ Dmytro Garastovych • a month ago

Hi. I have the same problem. I am trying to run it in existing project. I components I use styles, where I use

@import 'src/style/grid.scss';

and it fail to load it, throwing:

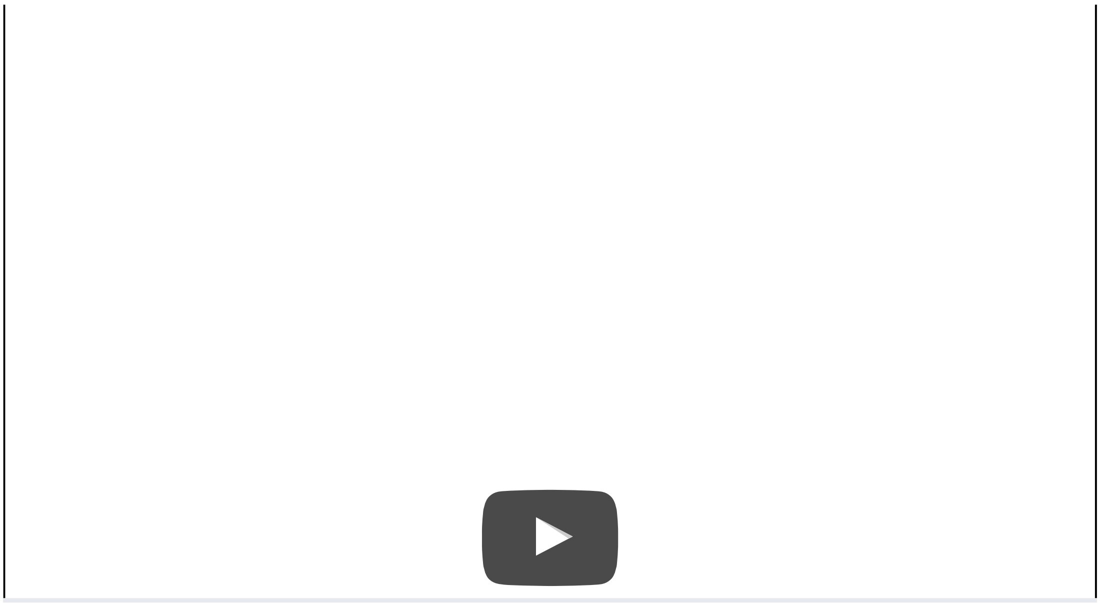
Error: Compilation failed. Resource file not found: /home/janusz/Projects/ui-project/src/app/shop/src/style/grid.scss
at ModuleResolutionHostAdapter.readResource (/home/janusz/Projects/ui-project/node_modules/@angular/compiler-cli/src/compiler_host.js:387:19)

⌃ | ⌄ • Reply • Share ›

Dmytro Garastovych ➦ Janusz • a month ago

Unfortunately, no one reads comments and replies here :(Yesterday fond a fresh video on youtube, but haven't checked it yet. Maybe it will help:

Angular Universal Part 2 - Building an app with Angular Universal

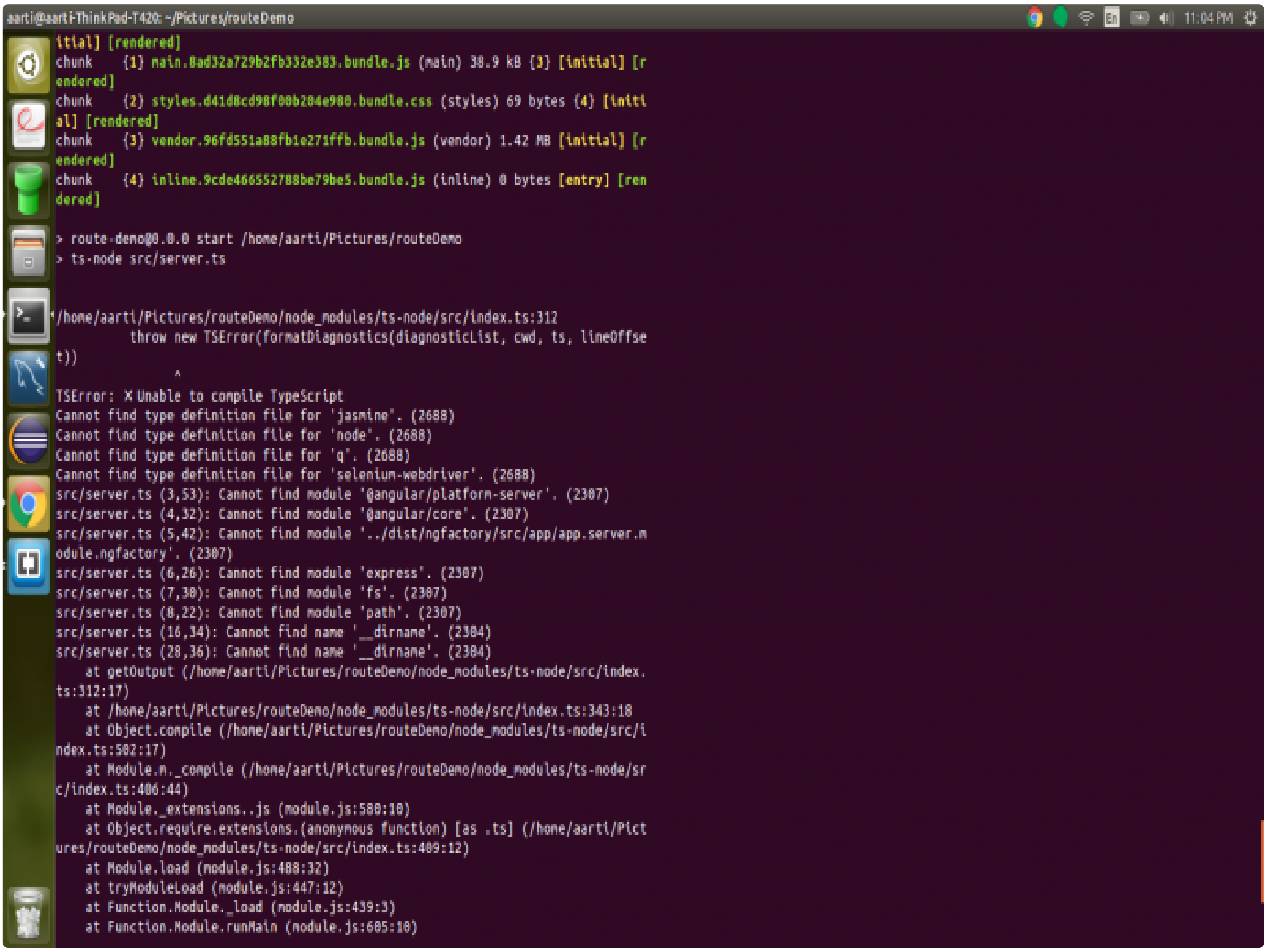


see more

^ | ⌄ • Reply • Share ›

Aarti Vadher • 17 days ago

trying to implement that into my project. But when running npm start run, i'm getting this error : any idea?



```
itial] [rendered]
chunk    {1} main.8ad32a729b2fb332e383.bundle.js (main) 38.9 kB {3} [initial] [r
endered]
chunk    {2} styles.d41d8cd98f00b204e980.bundle.css (styles) 69 bytes {4} [initi
al] [rendered]
chunk    {3} vendor.96fd551a88fb1e271ffb.bundle.js (vendor) 1.42 MB [initial] [r
endered]
chunk    {4} inline.9cde466552788be79be5.bundle.js (inline) 0 bytes [entry] [ren
dered]

> route-demo@0.0.0 start /home/aarti/Pictures/routeDemo
> ts-node src/server.ts


/home/aarti/Pictures/routeDemo/node_modules/ts-node/src/index.ts:312
          throw new TSError(formatDiagnostics(diagnosticList, cwd, ts, lineOffse
t))
                ^
TSError: ✗ Unable to compile TypeScript
Cannot find type definition file for 'jasmine'. (2688)
Cannot find type definition file for 'node'. (2688)
Cannot find type definition file for 'q'. (2688)
Cannot find type definition file for 'selenium-webdriver'. (2688)
src/server.ts (3,53): Cannot find module '@angular/platform-server'. (2307)
src/server.ts (4,32): Cannot find module '@angular/core'. (2307)
src/server.ts (5,42): Cannot find module '../dist/ngfactory/src/app/app.server.m
odule.ngfactory'. (2307)
src/server.ts (6,26): Cannot find module 'express'. (2307)
src/server.ts (7,30): Cannot find module 'fs'. (2307)
src/server.ts (8,22): Cannot find module 'path'. (2307)
src/server.ts (16,34): Cannot find name '__dirname'. (2304)
src/server.ts (28,36): Cannot find name '__dirname'. (2304)
    at getOutput (/home/aarti/Pictures/routeDemo/node_modules/ts-node/src/index.
ts:312:17)
    at /home/aarti/Pictures/routeDemo/node_modules/ts-node/src/index.ts:343:18
    at Object.compile (/home/aarti/Pictures/routeDemo/node_modules/ts-node/src/i
ndex.ts:502:17)
    at Module.m._compile (/home/aarti/Pictures/routeDemo/node_modules/ts-node/sr
c/index.ts:406:44)
    at Module._extensions..js (module.js:580:10)
    at Object.require.extensions.(anonymous function) [as .ts] (/home/aarti/Pict
ures/routeDemo/node_modules/ts-node/src/index.ts:409:12)
    at Module.load (module.js:488:32)
    at tryModuleLoad (module.js:447:12)
    at Function.Module._load (module.js:439:3)
    at Function.Module.runMain (module.js:605:10)
```

^ | ⌄ • Reply • Share ›

Ehsan Ershadi • 19 days ago

I'll get this error, any idea why?

```
> npm run start

> ang4-seo@0.0.0 prestart E:\Projects\Exams\ang4-seo
> ng build --prod && ngc

Hash: 2bb12db0f2570d3db6a5
Time: 14780ms
chunk {0} polyfills.106c1049bbf8c119e10f.bundle.js (polyfills) 160 kB {4} [initial] [rendered]
chunk {1} main.7384b4133e48ac7cb1f4.bundle.js (main) 16.5 kB {3} [initial] [rendered]
chunk {2} styles.d41d8cd98f00b204e980.bundle.css (styles) 69 bytes {4} [initial] [rendered]
chunk {3} vendor.65ec434f152c96fe621a.bundle.js (vendor) 1.13 MB [initial] [rendered]
chunk {4} inline.fe83fa25ffa9652554e8.bundle.js (inline) 0 bytes [entry] [rendered]
Error Unknown compiler option 'angularCompilerOptions'.

npm ERR! Windows_NT 10.0.14393
npm ERR! argv "C:\\Program Files\\nodejs\\node.exe" "C:\\Program
Files\\nodejs\\node_modules\\npm\\bin\\npm-cli.js" "ru
```

see more

Reply • Share ›

Ehsan Ershadi ➦ Ehsan Ershadi • 19 days ago

Ok, In windows 10 it took me a while to get it working, instead of 'npm run start' use these:

```
> ng build --prod
> .\node_modules\.bin\ngc -p tsconfig.json
> ts-node src/server.ts
```

Reply • Share ›

pablo nieto • a month ago

Thank you so much for this tutorial. how can we deploy this on server?

Reply • Share ›

pablo nieto • a month ago

thank you very much for this tutorial. How can I deploy this solution on server?

Reply • Share ›

Sam toms • a month ago

Hi Chaps, I have been part of the beta for a pre-render hosting service which means zero config for angular universal!!!!!

Just build with angular cli and deploy using roast. CDN, Amazon servers, redirects, and also a proxy for using external API's. Sit back put your feet up and hit deploy.

https://www.roast.io/

Reply • Share ›

Marian Stoica • a month ago

Great stuff! Finally, something concrete!
One thing missing though, in order to be able to use SSR: I quickly added a lazy module. When loading the app on that route, the content isn't rendered server side.

loading the app on that route, the content isn't rendered server-side.

⌃ | ⌄ • Reply • Share ›

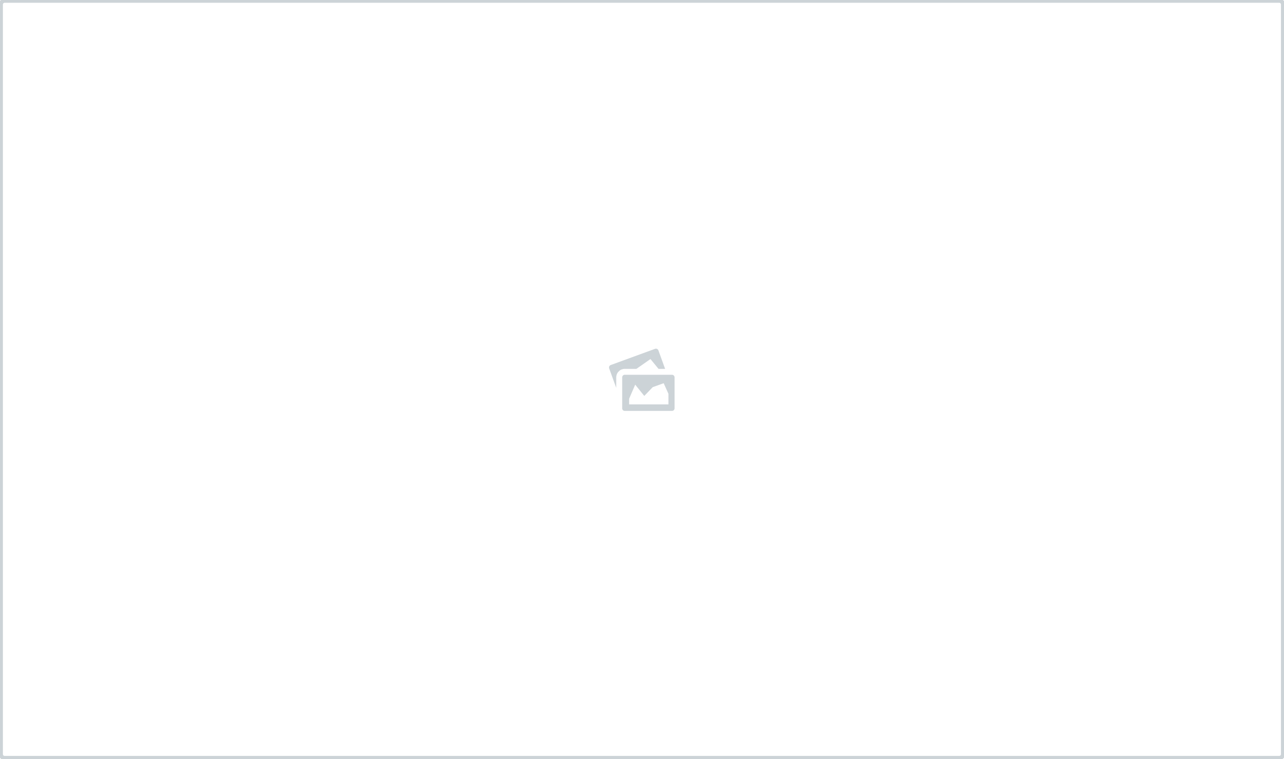
Can Kattwinkel ➦ Marian Stoica • a month ago

Yeah couldn't get it to work either. I'm facing a SystemJS Error on serverside..

⌃ | ⌄ • Reply • Share ›

Daniel Acosta • a month ago

Hello, could you help me I get this error and I do not know what it is?



⌃ | ⌄ • Reply • Share ›

amit chandra • 2 months ago

Thank You so much for giving a amazing tutorial.

⌃ | ⌄ • Reply • Share ›

Chris Rodriguez • 2 months ago

Getting the following error.

ERROR in H:/d/src/$$_gendir/app/app.server.module.ngfactory.ts (212,39): Namespace '"H:/d/node_modules/@angular/animations/browser"' has no exported member '?NoopAnimationEngine'.

⌃ | ⌄ • Reply • Share ›

Hemant Singh • 2 months ago

seriously awesome tutorial, love it , watched multiple universal tutorial wasted lot of time and was not able to create sample universal app with CLI, this is just latest and working love it, short, simple, working you rocks, Just taken latest of Angular (4.1.3), Bootstrap 4, and Angular CLI and it all worked here is the working repo https://github.com/hs950559...

⌃ | ⌄ • Reply • Share ›

realavaloro • 2 months ago

It's all good for me until I try to use the environment to read some properties.
It works well when rendered at client-side but when rendered at server-side with your server.ts I am getting an error.
See https://stackoverflow.com/q...

```
Error: Cannot find module 'environments/environment'
at Function.Module._resolveFilename (module.js:470:15)
at Function.Module._load (module.js:418:25)
at Module.require (module.js:498:17)
at require (internal/module.js:20:19)
at Object.<anonymous> (C:\Dev\code-carama\src\app\auth.service.ts:3:1)
at Module._compile (module.js:571:32)
at Module.m._compile (C:\Dev\code-carama\node_modules\ts-node\src\index.ts:385:23)
at Module._extensions..js (module.js:580:10)
at Object.require.extensions.(anonymous function) [as .ts] (C:\Dev\code-
carama\node_modules\ts-node\src\index.ts:388:12)
at Module.load (module.js:488:32)
```

see more

^ | ˅ • Reply • Share ›

realavaloro ➦ realavaloro • 2 months ago

Never mind, it seems the problem is that although for client side rendering the line:
import { environment } from 'environments/environment';
works well, for server side rendering the line must be:
import { environment } from '../environments/environment';

and by the way, I don't know why but by default for server-side rendering it will pick the PROD environment file..

^ | ˅ • Reply • Share ›

realavaloro • 2 months ago

I have tried it and it works fine with server side rendering. Thank you!
However I am having issues with SOME of the content that should be rendered at server-side not doing so and instead it's being rendered at client-side and I am not sure why.
I have described the issue here: https://stackoverflow.com/q...

Basically I am trying to have knockoutJs within Angular 4 because I need to render at server side some html that has knockout binding attributes, so my hopes was for Angular Universal to WAIT for the knockout to apply bindings and then render the html in server side..
Thanks again for any advice!

^ | ˅ • Reply • Share ›

salman • 2 months ago

Hi there, can you please tell me How to Deploy this , i have a VPS and node is running on server, Can you please guid me how i would deploy this , do i have to make an express server on server first or what?

^ | ˅ • Reply • Share ›

Michalis Tzikas ➦ salman • 2 months ago

I have the same question with you... Did you find out what we have to do to work in production?

^ | ˅ • Reply • Share ›

Yakov Fain • 2 months ago

Good article, thanks!
A small correction: instead of "npm run start" you can do "npm start".

HOME

COURSES

CHALLENGES

LOGIN

JOIN

TUTORIALS

CONTACT US

CHAT WITH US