

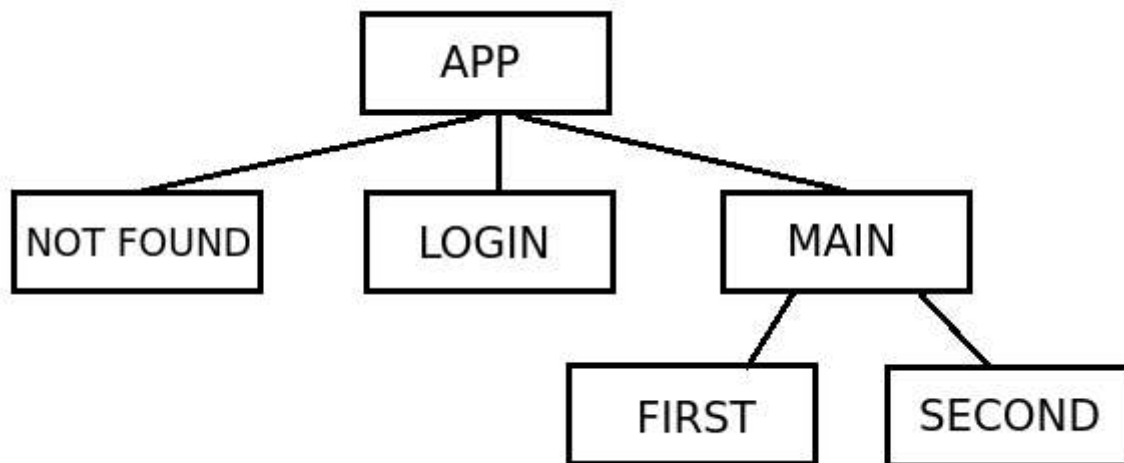
Angular2 router: how to correctly load children modules with their own routing rules

here is my Angular2 app structure:

31



7



Here is part of my code. The following is the main module of the Angular2 app, that imports its routing rules and a child module (EdgeModule) and uses some components related to some pages.

app.module.ts

```
@NgModule({
  declarations: [
    AppComponent,
    PageNotFoundComponent,
    LoginComponent
  ],
  imports: [
```

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



```

    ],
    bootstrap: [AppComponent]
  })

  export class AppModule {
  }

```

Here is the routing rules for the main module. It have paths to login page and page not found.

app.routing.ts

```

const appRoutes: Routes = [
  { path: 'login', component: LoginComponent },
  { path: '**', component: PageNotFoundComponent }
];

export const appRoutingProviders: any[] = [];

export const appRouting = RouterModule.forRoot(appRoutes, { useHash: true });

```

Here is `EdgeModule` that declares the component that it uses and import its own routing rules and 2 child modules (`FirstSectionModule` and `SecondSectionModule`).

edge.module.ts

```

@NgModule({
  declarations: [
    EdgeComponent,
    SidebarComponent,
    TopbarComponent
  ],
  imports: [
    ...
    edgeRouting,
    FirstSectionModule,
    SecondSectionModule
  ],
  providers: [
    AuthGuard
  ]
})

```

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

 Google

Facebook 

Here is the routing rules for the module that loads, as you can see, topbar and sidebar components.

edge.routing.ts

```
Paths['edgePaths'] = {
  firstSection: 'firstSection',
  secondSection: 'secondSection'
};

const appRoutes: Routes = [
  { path: '', component: EdgeComponent,
    canActivate: [AuthGuard],
    children: [
      { path: Paths.edgePaths.firstSection, loadChildren: '../somepath/first-
section.module#FirstModule' },
      { path: Paths.edgePaths.secondSection, loadChildren:
'../someotherpath/second-section.module#SecondModule' },
      { path: '', redirectTo: edgePaths.dashboard, pathMatch: 'full' }
    ]
  }
];

export const edgeRouting = RouterModule.forChild(appRoutes);
```

Finally, this is one of the two child module, that have its components and imports its routing rules.

first-section.module.ts

```
@NgModule({
  declarations: [
    FirstSectionComponent,
    SomeComponent
  ],
  imports: [
    ...
    firstSectionRouting
  ],
  providers: [
    AuthGuard,
  ],
})
```

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

 Google

Facebook 

These are the routing rules for the pages (components) of `FirstSectionModule`

`first-section.routing.ts`

```
Paths['firstSectionPaths'] = {
  someSubPage: 'some-sub-page',
  someOtherSubPage: 'some-other-sub-page'
};

const appRoutes: Routes = [
  {
    path: '',
    children: [
      { path: Paths.firstSectionPaths.someSubPage, component:
someSubPageComponent},
      { path: Paths.firstSectionPaths.someOtherSubPage, component:
someOtherSubPageComponent},
      { path: '', component: AnagraficheComponent }
    ]
  }
];

export const firstSectionRouting = RouterModule.forChild(appRoutes);
```

Almost the same happens for `second-section.module.ts` and `second-section.routing.ts` files.

When i run the app the first things that load is the page related to `FirstSectionComponent`, with no sidebar nor topbar.

Can you tell me what's wrong with my code? There are not errors in the console.



asked Oct 18 '16 at 14:22



[smartmouse](#)

4,849 17 60 113

1 [trv removing FirstSectionModule and SecondSectionModule from edgeModule import](#) — [Gaurav Mukherjee](#) Oct 18 '16 at 14:54

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



19 '16 at 10:32

- 1 I did not find something similar to 'loadChildren' that could inform the framework to 'plug' a child module route config at a specific point ... – Clement Jan 18 '17 at 15:21

4 Answers

▲ You can try this using `loadChildren` where the `homeModule` , `productModule` , `aboutModule` have their own route rules.

8



```
const routes: Routes = [
  { path: 'home', loadChildren: 'app/areas/home/home.module#homeModule' },
  { path: 'product', loadChildren: 'app/areas/product/product.module#ProductModule' },
  { path: 'drawing', loadChildren: 'app/areas/about/about.module#AboutModule' }
];

export const appRouting = RouterModule.forRoot(routes);
```

and the home route rules will be like

```
export const RouteConfig: Routes = [
  {
    path: '',
    component: HomeComponent,
    canActivate: [AuthGuard],
    children: [
      { path: '', component: HomePage },
      { path: 'test/:id', component: Testinfo},
      { path: 'test2/:id', component: Testinfo1},
      { path: 'test3/:id', component: Testinfo2}
    ]
  }
];
```

this is also known as lazy loading the modules.

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



nothing special about `LazyModule` other than it has its own routing and a component called `LazyComponent` .

Check out this awesome tutorial related to this: <https://angular-2-training-book.rangle.io/handout/modules/lazy-loading-module.html>

edited Mar 18 at 12:02



g t

4,654

4

38

80

answered Jun 7 '17 at 3:23



Yashwanth Chowdary

Kata

424

6

12

Thank you for your answer, anyway the problem with lazy loading ans SystemJS is that these paths doesn't exist when you build your app for production, when you get one minified JS file as output of Gulp's build task. – [smartmouse](#) Jun 7 '17 at 11:56

did you try adding `<base href="/">` in the index.html – [Yashwanth Chowdary Kata](#) Jun 7 '17 at 12:17

I mean that in the minified JS file I have not references to paths, such as `/lazy/lazy.module` – [smartmouse](#) Jun 7 '17 at 12:49



1



In your `app.routing.ts`, there are only 2 routes and no route included to navigate to the Main section (as in the diagram). There needs to be a route entry with `loadchildren` property so it will load the module for the Main section.

```
routes: Routes = [...  
{  
  path: 'main', loadChildren: '<file path>/<Edge module file name>#EdgeModule'  
},  
...];
```

This will load the rest of the modules, components routes and everything insite the `EdgeModule`.

answered Aug 9 '17 at 11:54



CharithW

21

5

Not sure if I got the problem correctly, but here is a small code snippet which I used to generate routes dynamically.

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



```
constructor(private _router: Router) {  
}  
  
ngOnInit() {  
  ...  
  this._router.config[0].children = myService.getRoutes();  
  this._router.resetConfig(this._router.config);  
  console.debug('Routes:', this._router.config);  
  ...  
}
```

It is not OOTB solution, but you can get information about current routes.

answered Feb 3 '17 at 16:24



[Sargon](#)

223 1 8

Could you be more specific? – [alexndreazevedo](#) Feb 9 '17 at 19:59

This is absolutely not a good practice, don't do that please – [SherloxFR](#) May 9 '17 at 15:01

ok, show me please how to generate routes dynamically, after application started. – [Sargon](#) May 12 '17 at 22:43



0



It's a dependency injection problem. We don't need to inject FirstSectionModule & SecondSectionModule in the edgeModule & about route we can use inside of FirstSectionModule & SecondSectionModule. So just removing it from edgeModule will work.

answered Jul 19 '17 at 3:18



[Vivek Kumar](#)

46 3

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

