Angular 2 routing redirect to with child routes

25

6

I am a newbie in Angular 2. I want to create isolated modules for every part of my app. For example I created the AuthModule with default component - AuthComponent which contain a router outlet for his child components (SignIn or SignUp). So I want to realize the following scenario:

1. When navigate to / - root off app - redirect to /auth
2. After redirect to /auth - load AuthComponent with router outlet
3. After AppComponent loaded - load default sign-in component via redirecting to /auth/sign-in

But when I going to localhost/ I get redirect to /auth what I want, but the next redirect to sign-in doesn't appear.

My code: app.routing

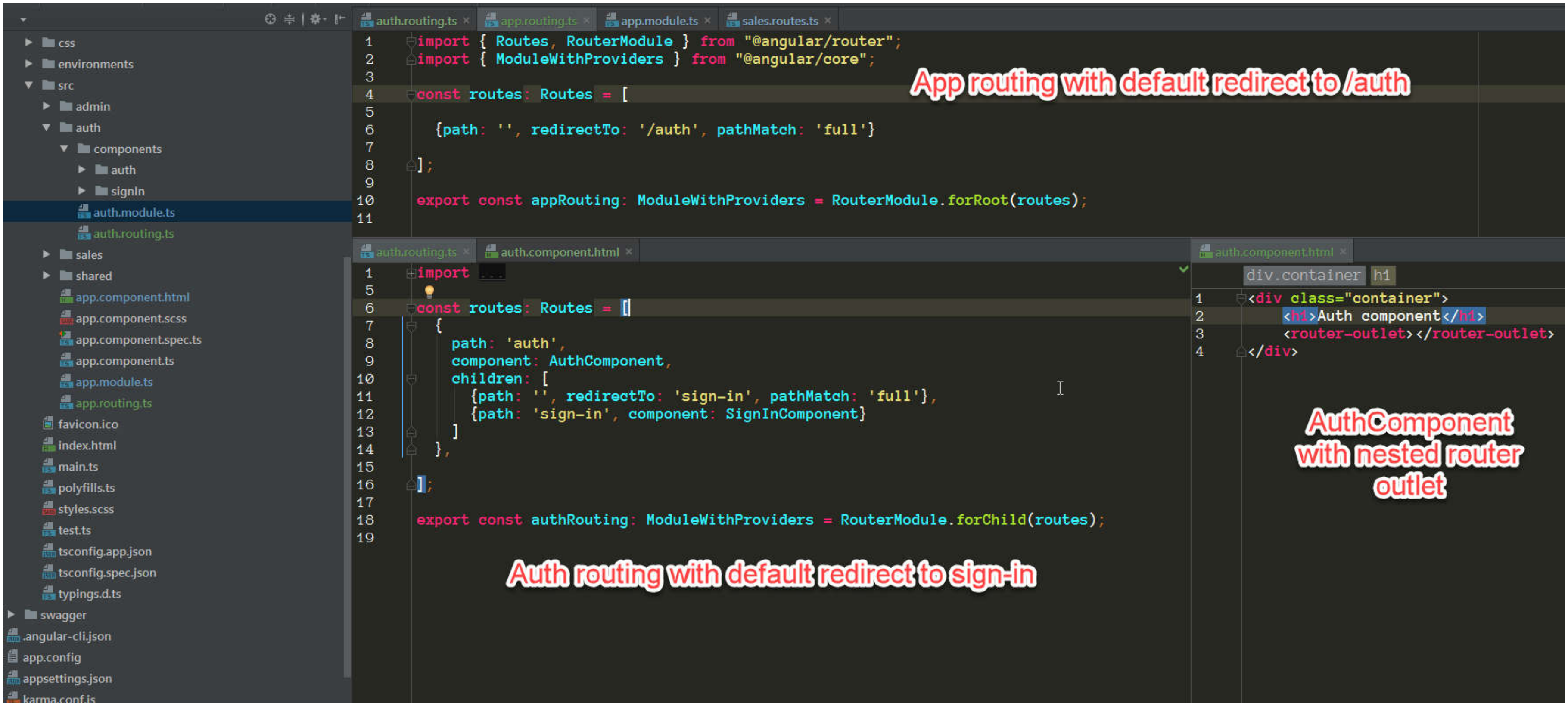
```
const routes: Routes = [
  {path: '', redirectTo: '/auth', pathMatch: 'full'}
];
export const appRouting: ModuleWithProviders = RouterModule.forRoot(routes);
```

auth.routing

```
const routes: Routes = [
  {
    path: 'auth',
    component: AuthComponent,
    children: [
      {path: '', redirectTo: 'sign-in', pathMatch: 'full'},
      {path: 'sign-in', component: SignInComponent}
    ]
  },

];
```

```
<div class="container">
    <h1>Auth component</h1>
    <router-outlet></router-outlet>
</div>
```

Result:



```
import { Routes, RouterModule } from "@angular/router";
import { ModuleWithProviders } from "@angular/core";

const routes: Routes = [

  {path: '', redirectTo: '/auth', pathMatch: 'full'}

];

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

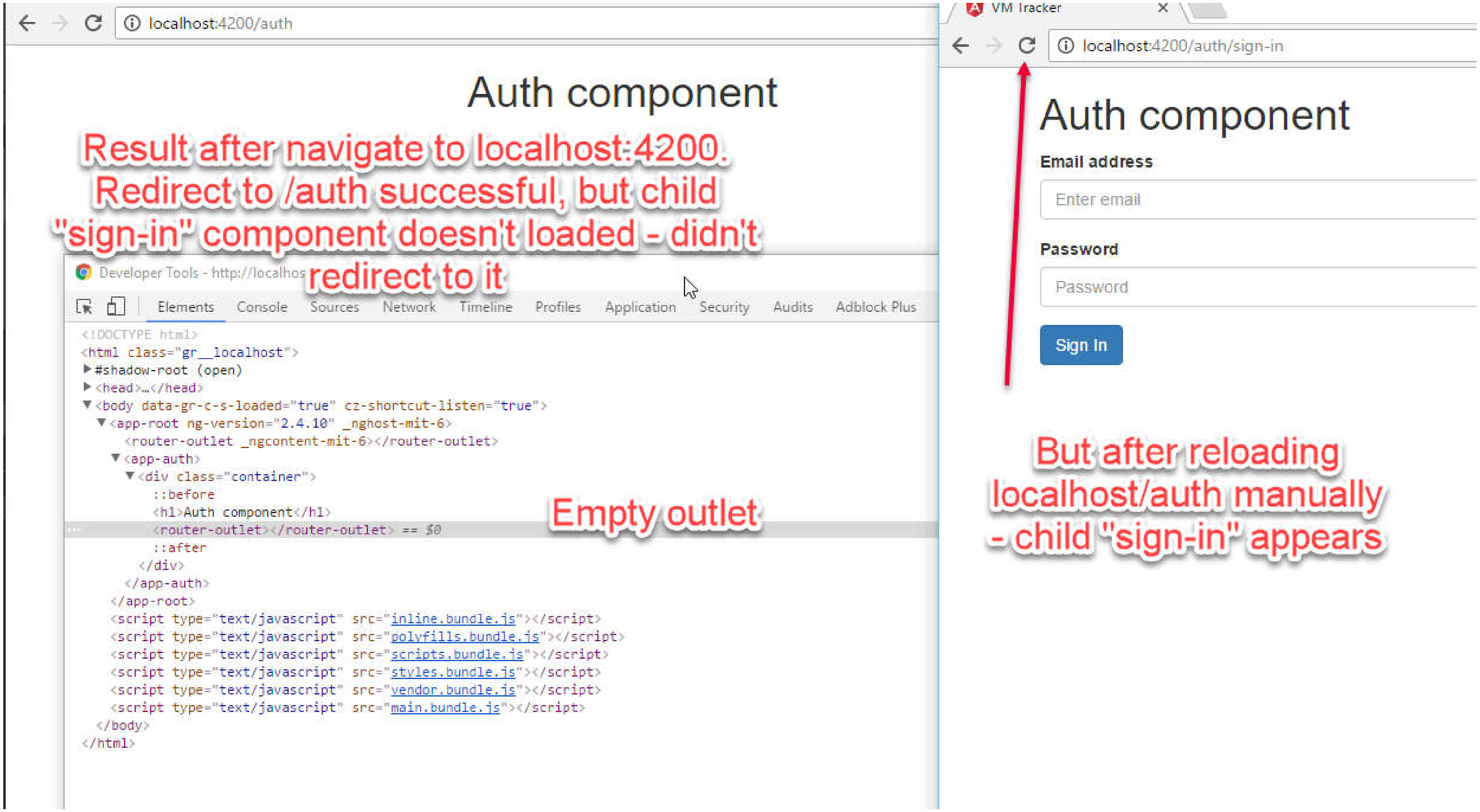
App routing with default redirect to /auth

```
import ...

const routes: Routes = [
  {
    path: 'auth',
    component: AuthComponent,
    children: [
      {path: '', redirectTo: 'sign-in', pathMatch: 'full'},
      {path: 'sign-in', component: SignInComponent}
    ]
  },

];

export const authRouting: ModuleWithProviders = RouterModule.forChild(routes);
```

Auth routing with default redirect to sign-in

```
<div class="container">
    <h1>Auth component</h1>
    <router-outlet></router-outlet>
</div>
```

AuthComponent with nested router outlet



Auth component

Result after navigate to localhost:4200. Redirect to /auth successful, but child "sign-in" component doesn't loaded - didn't redirect to it

```
<!DOCTYPE html>
<html class="gr__localhost">
▶#shadow-root (open)
▶<head>…</head>
▼<body data-gr-c-s-loaded="true" cz-shortcut-listen="true">
  ▼<app-root ng-version="2.4.10" _nghost-mit-6>
      <router-outlet _ngcontent-mit-6></router-outlet>
    ▼<app-auth>
      ▼<div class="container">
          ::before
          <h1>Auth component</h1>
          <router-outlet></router-outlet> == $0
          ::after
        </div>
      </app-auth>
    </app-root>
    <script type="text/javascript" src="inline.bundle.js"></script>
    <script type="text/javascript" src="polyfills.bundle.js"></script>
    <script type="text/javascript" src="scripts.bundle.js"></script>
    <script type="text/javascript" src="styles.bundle.js"></script>
    <script type="text/javascript" src="vendor.bundle.js"></script>
    <script type="text/javascript" src="main.bundle.js"></script>
  </body>
</html>
```

Empty outlet

Auth component

But after reloading localhost/auth manually - child "sign-in" appears

Environment @angular/cli: 1.0.0-rc.2 node: 7.7.1 os: win32 x64

angular routing children

edited Mar 18 '17 at 13:33

asked Mar 18 '17 at 13:14

Timofev Orischenko

7 Answers

28

I have been the same problem. It seems an Angular tricks: If you remove leading slash in 'redirectTo' field, your application will be redirected successfully to auth/sign-in.

Use this in app.routing:

```
const routes: Routes = [

    {path: '', redirectTo: 'auth', pathMatch: 'full'},

];
```

‘redirectTo’ value starts with a ‘/’ = absolute path

‘redirectTo’ value starts without a ‘/’ = relative path

Read more about it: https://vsavkin.com/angular-router-understanding-redirects-2826177761fc

P.S My opinion that your structure more correctly then YounesM's one. Parent module can't keep children routes: "app" module don't know that "auth" module have children module "sign-in".

answered May 2 '17 at 11:46
Dmitriy Ivanko
331 2 10

So, what it seems to happen is that when you `redirectTo:'auth'` it tries to load the `''` children's component and since it does not have any component the router-outlet is empty

```
const routes: Routes = [
  {path: '', redirectTo: '/auth/sign-in', pathMatch: 'full'}
];
export const appRouting: ModuleWithProviders = RouterModule.forRoot(routes);
```

auth.routes

```
const routes: Routes = [
  {
    path: 'auth',
    component: AuthComponent,
    children: [
      {path: 'sign-in', component: SignInComponent}
    ]
  },

];
```

or have a component in your `''` path instead of redirecting.

app.routes

```
const routes: Routes = [
  {path: '', redirectTo: '/auth', pathMatch: 'full'}
];
export const appRouting: ModuleWithProviders = RouterModule.forRoot(routes);
```

auth.routes

```
const routes: Routes = [
  {
    path: 'auth',
    component: AuthComponent,
    children: [
      {path: '', component: SignInComponent}
    ]
  }
```

Kabb5
2,747 2 25 46

YounesM
1,722 9 23

Thanks for the answer. Yes, I can redirect to /auth/sign-in from app routes directly and it works perfectly. But I don't want to open internal structure of AuthModule (/auth/**sign-in** part of route, for example) You said: *So, what it seems to happen is that when you redirectTo:'auth' it tries to load the '' children's component and since it does not have any component the router-outlet is empty.* but I have a default children router – Timofey Orischenko Mar 18 '17 at 15:23

As long as you use `''` route to redirect it seems that your `routerLink` s will not work properly (since they're trying to load the component) the other alternative will be to have a component to load in the `''` route (as edited). I don't see another way to get around it. – YounesM Mar 18 '17 at 15:33

Yes, it will work, but in this case we have no possibility to change url to /auth/ **sign-in** I thought there is a best practice to organize routes inside modules in angular 2. Thanks for your answer – Timofey Orischenko Mar 18 '17 at 16:06

1

It's quite simple:

```
const routes: Routes = [
    { path: '', redirectTo: '/auth/signin', pathMatch: 'full' },
    { path: 'auth', component: AuthComponent,
        children: [
            { path: 'signup', component: SignupComponent },
            { path: 'signin', component: SigninComponent },
            { path: 'logout', component: LogoutComponent }
        ]
    },
    { path: '**', redirectTo: '/auth/signin', pathMatch: 'full' }
];
```

edited Aug 2 '17 at 23:32
Anh Pham
1,719 6 13 23

answered Aug 2 '17 at 17:47
user2912589

You can use the re-direction in your outer module, like this:

```
];

// Child routing module or child.routing.module
const routes: Routes = [
 {
   path: 'auth',
   //component: AuthComponent, may not need this as you only need the template
   children: [
     {path: 'sign-in', component: SignInComponent}
   ]
 },
];
```

You need not to have an empty path '' in your child component in case you want to navigate to this with a proper path

answered Mar 20 '18 at 2:55

bitscanbyte
295 2 9

1

on auth.routes

```
const routes: Routes = [
{
        path: 'auth',
        redirectTo: 'auth/sign-in'
  },
{
    path: 'auth',
    component: AuthComponent,
    children: [
      {path: 'sign-in', component: SignInComponent}
    ]
  },

];
```

0

You can do this in your child route:

```
const routes: Routes = [
  { path: 'auth', redirectTo: 'auth/signin'},
  {
    path: 'auth',
    component: AuthComponent,
    children: [{ path: 'signin', component: SignInComponent }],
  },
];
```

answered Jul 20 '18 at 12:13

papiliond

1 1

0

I have been the same problem, but none of the above answers seems to be a good resolution. In my code , i subscribe to the router events and finally resolve it.

code in AuthComponent's constructor:

```
  this.router.events.subscribe((e: Event) => {
   if (e instanceof NavigationEnd) {
    this.activeUrl = e.urlAfterRedirects || e.url;
    if (this.activeUrl === '/auth') {
      this.router.navigateByUrl('/auth/sign-in');
    }
   }
  });
```

answered Sep 3 '18 at 7:10

johney

1