Configuring Routes in Angular

Introduction

Angular is a *single-page application* (SPA), which means it has only one path. For example, if you're running your application on localhost with a default port, then it would be http://localhost:4200.

But in real-life applications, there are multiple pages you want to show a user after some events or in the navigation. So how can we achieve these things in Angular?

That's where Angular router comes into the picture.

Intro to Angular Router

Angular router enables the user to navigate between different components after some events. It creates the experience of a multiple-page application in a single-page application.

Let's see how to configure the routes in Angular.

Create an Angular Project with cmd

Before diving deep into Angular Router, we need an Angular project. Let's make a project using CMD that will automatically add up the necessary things a project needs to run.

1

ng g new project-name

console

Now we're ready to go. We have all the necessary things to get started.

Routing Configurations

After creating the project, you can see a newly generated folder with the same name that you have given at the time of creating the project. In that folder, you can see the file /src/app/app-routing.module.ts.

This file is where we need to configure our router.

Initially, it would look like the below snippet.

1

2

3

4

5

6

7

8

9

10

import { NgModule } from "@angular/core";

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

const routes: Routes = [];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule {}

typescript

Here we need to configure routes with router module, with the help of RouterModule.forRoot() and imports in the @NgModule.

Angular router doesn't have any values until you configure it. So we need to add value in the array of routes.

Configure the Array of Routes

Routes array contains the object of Route. Route is an interface that provides many variables, but we'll not go into much detail for now.

A basic configuration mostly uses two variables: path and component.

path defines the URL path.

component specifies the component.

Let's make two different components: FirstComponent and SecondComponent, respectively.

In the below snippet, you'll find the command for creating components using cmd.

1

2

3

ng g c first --skipTests=true

ng g c second --skipTest=true

console

Now we need to configure the components with an object of Route interface.

The object structure will look as follows-

1

2

3

4

{

path:'pathString',

component:'componentname'

}

ts

So in the above snippet, I have given you a sample demo.

1

2

3

4

5

6

7

8

9

10

const routes: Routes = [

{

path: "first",

component: FirstComponent

},

{

path: "second",

component: SecondComponent

}

];

typescript

Here is a configured route. I've added two routes in the routes array.

Router Outlet

Now we need to add a <router-outlet > tag in app.component.html

app.component.html

1

<router-outlet></router-outlet>

html

We can access the configured path by visiting the same URL given in the path variable heading with your host and port.

In our example, these two URLs are:

1

2

3

http://localhost:4200/first

http://localhost:4200/second

console

Conclusion

Angular routing is handy to make an application that behaves like a multi-page application. You can add a lot more configuration in Angular.