

Module 7 – Angular Routes and Navigation

Demo Document 2 – Make use of route guards to prevent navigation to different pages

edureka!

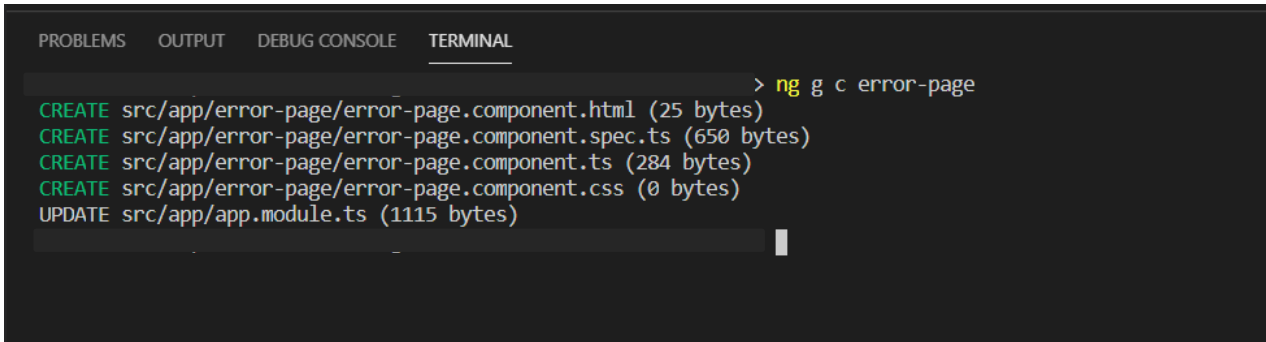
edureka!

© Brain4ce Education Solutions Pvt. Ltd.

In this demo, we will see how create different types of route e.g. nested and wild card routes and make use of route guards.

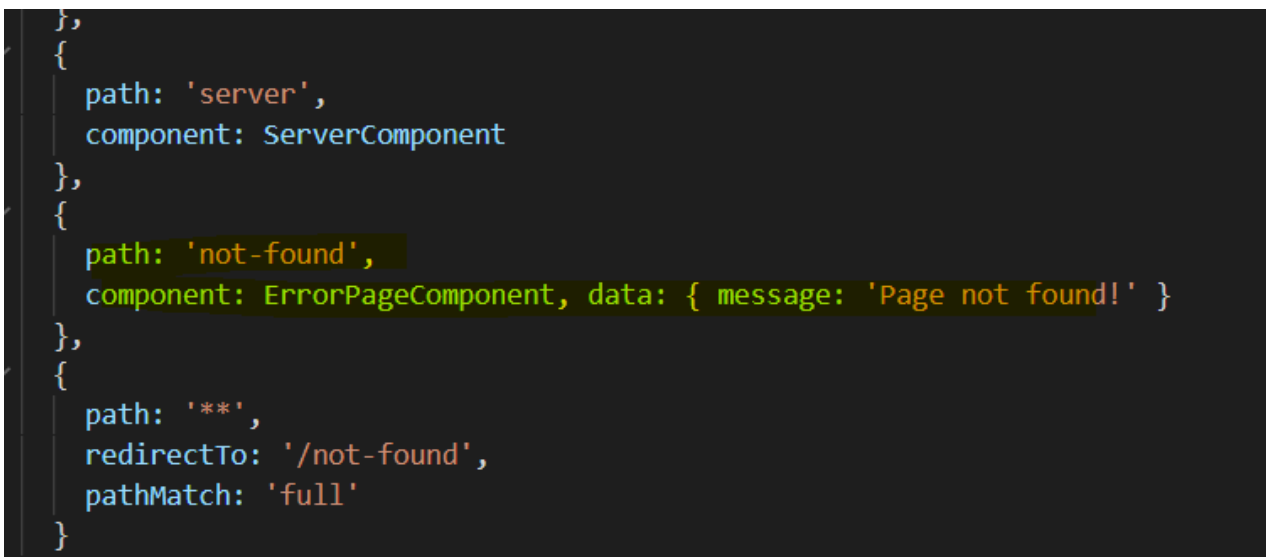
Step 1 – Open visual studio code, open same app created in demo1.

Create new component for error page using `ng g c error-page` command



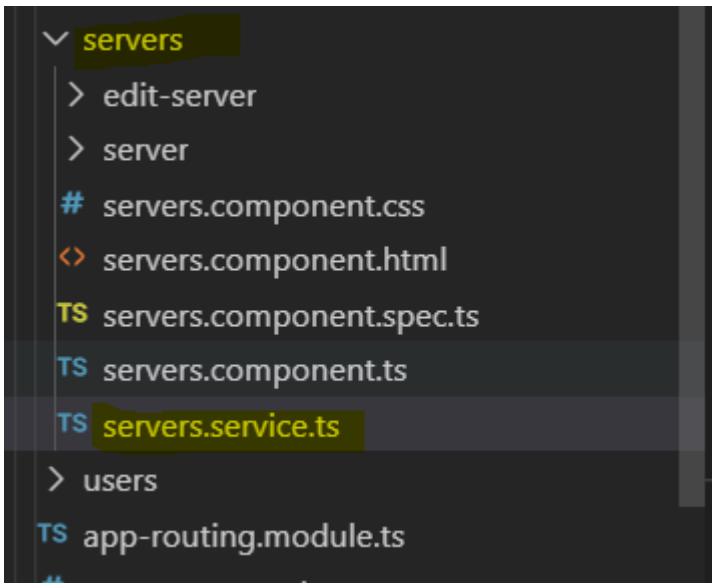
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
> ng g c error-page
CREATE src/app/error-page/error-page.component.html (25 bytes)
CREATE src/app/error-page/error-page.component.spec.ts (650 bytes)
CREATE src/app/error-page/error-page.component.ts (284 bytes)
CREATE src/app/error-page/error-page.component.css (0 bytes)
UPDATE src/app/app.module.ts (1115 bytes)
```

Step 3 –now open app-routing.module.ts file add below line of code ,



```
},
{
  path: 'server',
  component: ServerComponent
},
{
  path: 'not-found',
  component: ErrorPageComponent, data: { message: 'Page not found!' }
},
{
  path: '**',
  redirectTo: '/not-found',
  pathMatch: 'full'
}
```

step 4—now create servers.service.ts inside servers folder and copy below code



```
export class ServersService {
  private servers = [
    {
      id: 1,
      name: 'Productionserver',
      status: 'online'
    },
    {
      id: 2,
      name: 'Testserver',
      status: 'offline'
    },
    {
      id: 3,
      name: 'Devserver',
      status: 'offline'
    }
  ];

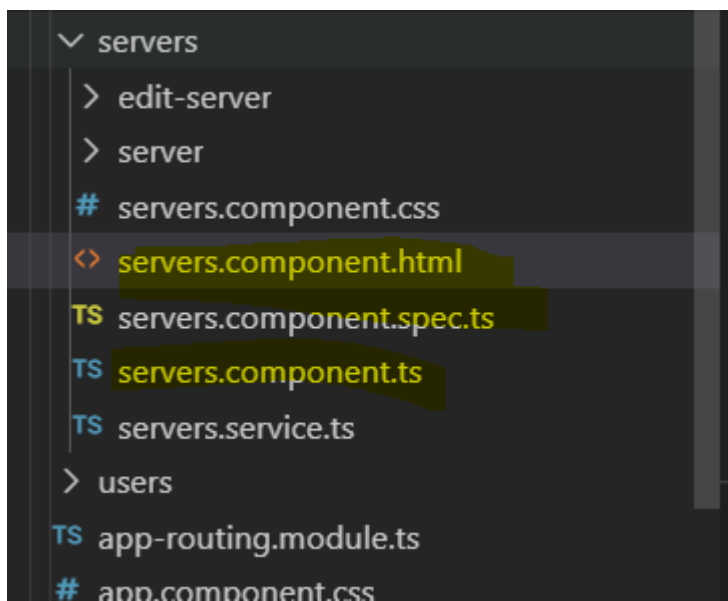
  getServers() {
    return this.servers;
  }

  getServer(id: number) {
    const server = this.servers.find(
      (s) => {
        return s.id === id;
      }
    );
    return server;
  }
}
```

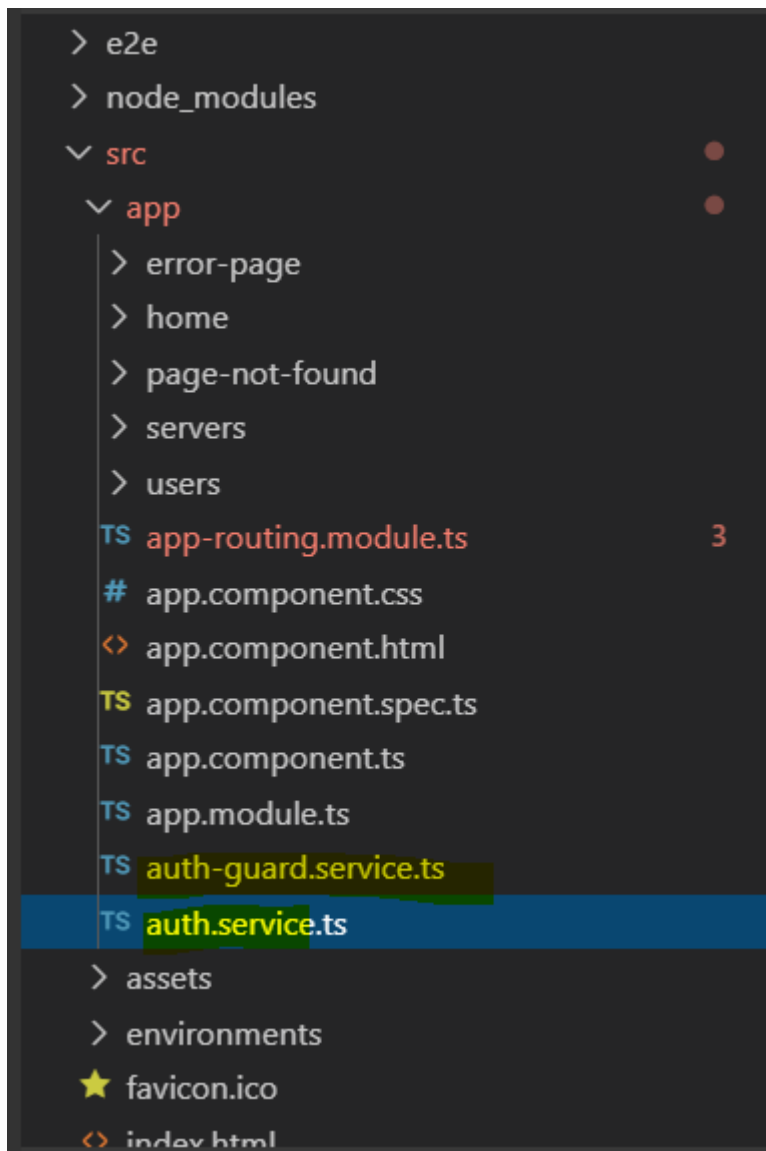
```
updateServer(id: number, serverInfo: {name: string, status: string}) {  
  const server = this.servers.find(  
    (s) => {  
      return s.id === id;  
    }  
  );  
  if (server) {  
    server.name = serverInfo.name;  
    server.status = serverInfo.status;  
  }  
}
```

This will create service for servers operations.

Step 5 – copy html and ts file code from demo folder to servres component.

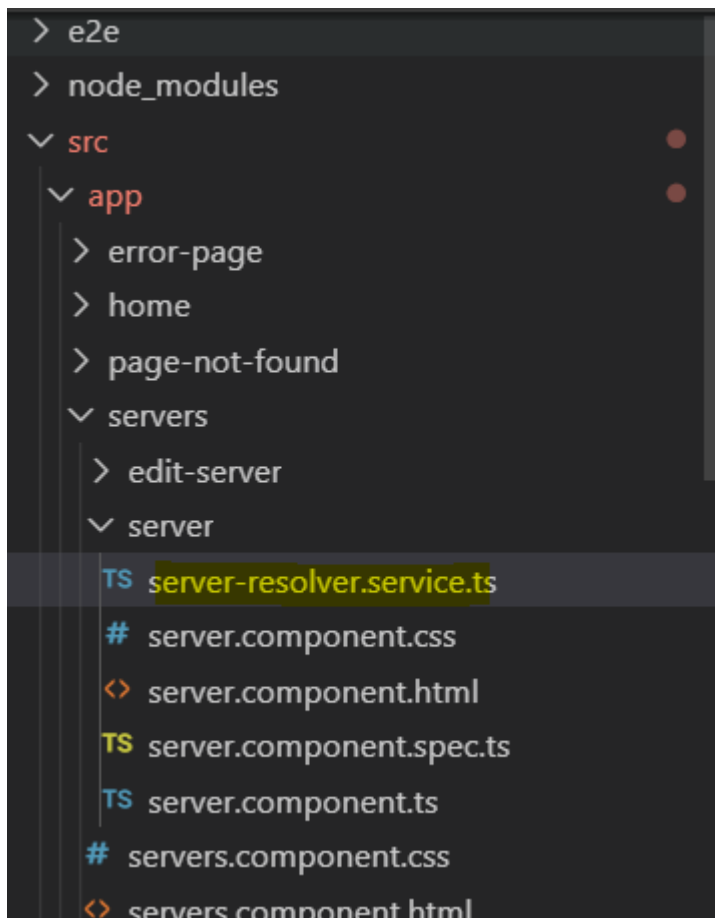


Step 6 – Create auth guard service as auth-guard.service.ts and auth.service.ts as below



Copy code from demo folder to both the above files created.

Step 7 – go to `src/app/servers/server` and create server resolver service as below



Copy code from demo folder

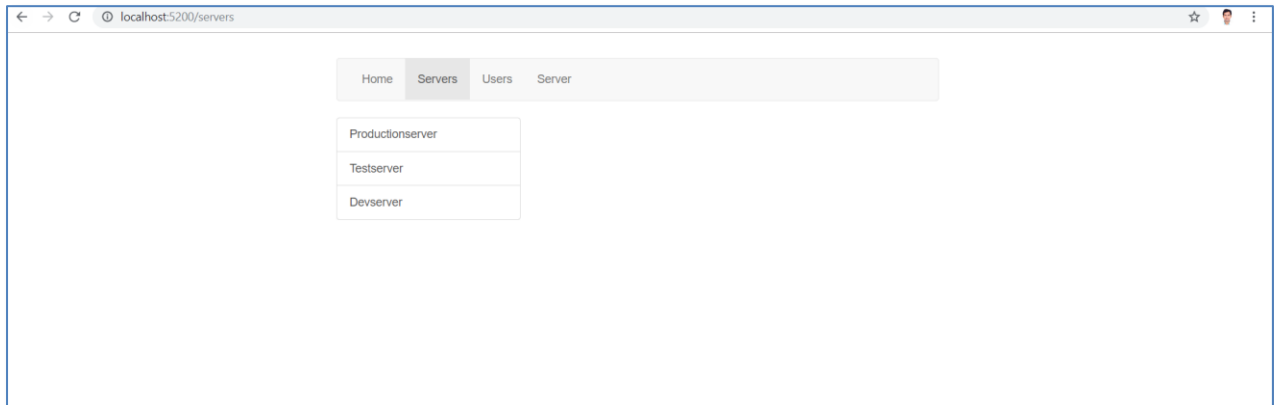
Step 8 – now open `app-routing.module.ts` and add below code

```
{
  path: 'servers',
  canActivateChild: [AuthGuard],
  component: ServersComponent,
  children: [
    {
      path: ':id',
      component: ServerComponent, resolve: { server: ServerResolver }
    },
    {
      path: ':id/edit',
      component: EditServerComponent, canActivate: [CanDeactivateGuard]
    }
  ]
},
```

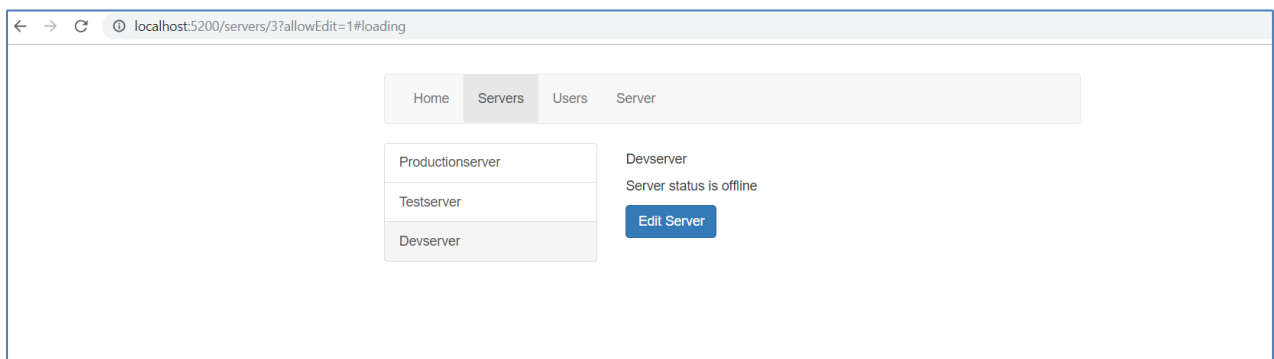
Step 9 – copy html and ts code from demo folder to edit-server and server components

Step 10 – run using ng serve

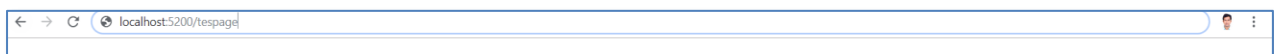
Click on servers



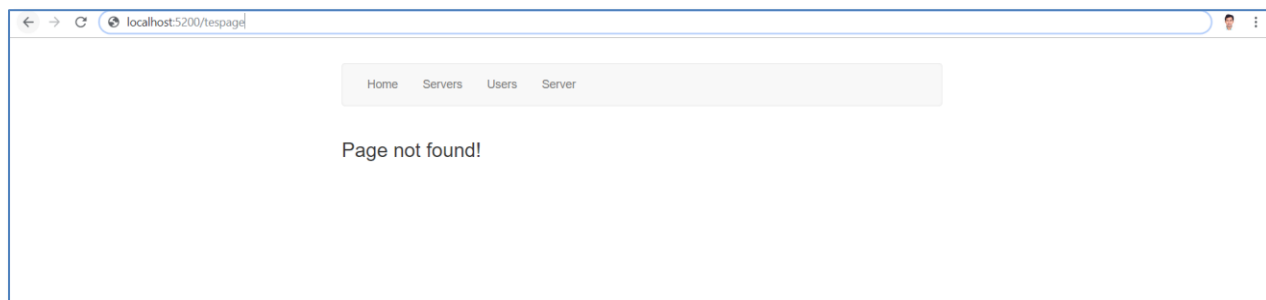
We can see different serves listed. Now click on any server, it will redirect to home page



Now we will see wild card routing, change url manually as below



Enter any invalid url and click enter. we will see page not found error



We have also used auth guard service to protect fake auth and auth service for login

edureka!