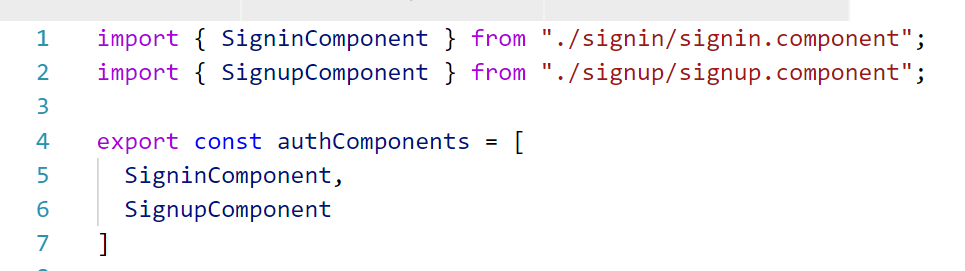
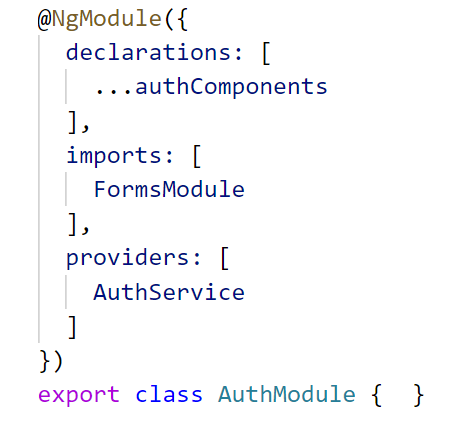
# Angular Furniture System - Refactoring Exercise

## Create Modules

It’s a good idea as the app grows to **add modules** for each part of our project structure. Create an **index.ts** file at **authentication** and inside import Sign In and Sign Up components and **export** them as an **array**:



After that create an **auth.module.ts** and inside define a module which declares the components using the **index.ts** file (use **destructuring assignment** inside declarations) and **provide** the authentication service:



Notice that we also need to import the **FormsModule** since we use it inside both form components.

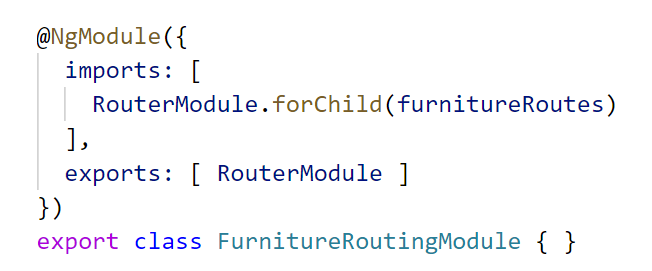
Now all we need to do is go to the **main app module** and import the newly created authentication module and remove everything else we no longer need in app module.

Create a **furniture module** in the same matter.

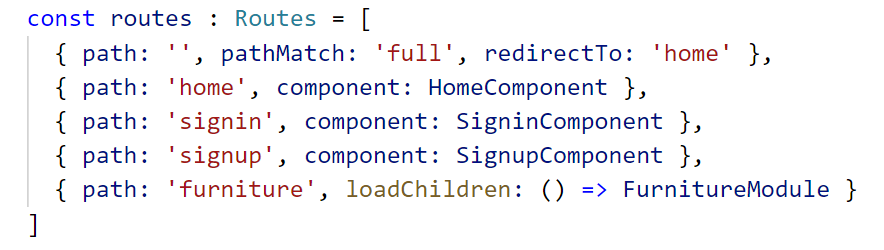
## Lazy Load Module

In order to **speed up** our application start time a bit and load a certain module only when we need it we have to implement lazy loading on this module.

Create a **furniture-routing.module.ts** and inside define all **child** furniture routes. Wrap them up in a **module** and use the **RouterModule.forChild()** function:



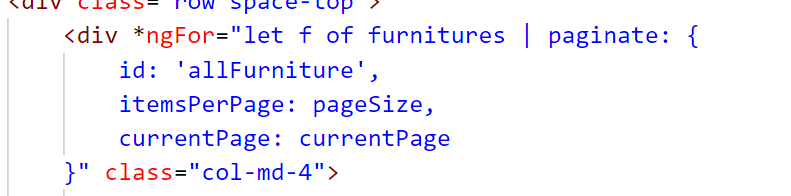
Import this module inside **furniture module**. Now go to the main **app routing module** and inside use the property **loadChildren** to load furniture module lazily:

****

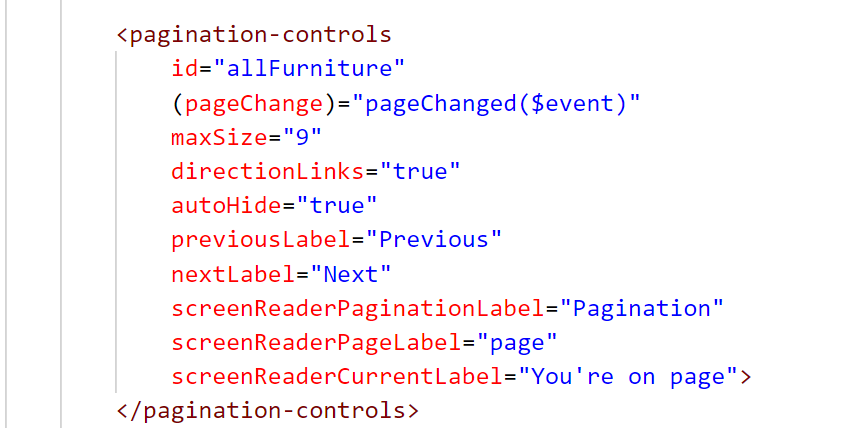
## Pagination

We can add client side pagination to our application. To achieve this we need to install **ngx-pagination** and import **NgxPaginationModule** inside **furniture module** (don’t forget to import **CommonModule** as well since it needs it).

To create pagination inside **furniture-all** component we need to use the **paginate pipe** and set an **id**, **pageSize** and **currentPage** (both of which are defined inside the component):



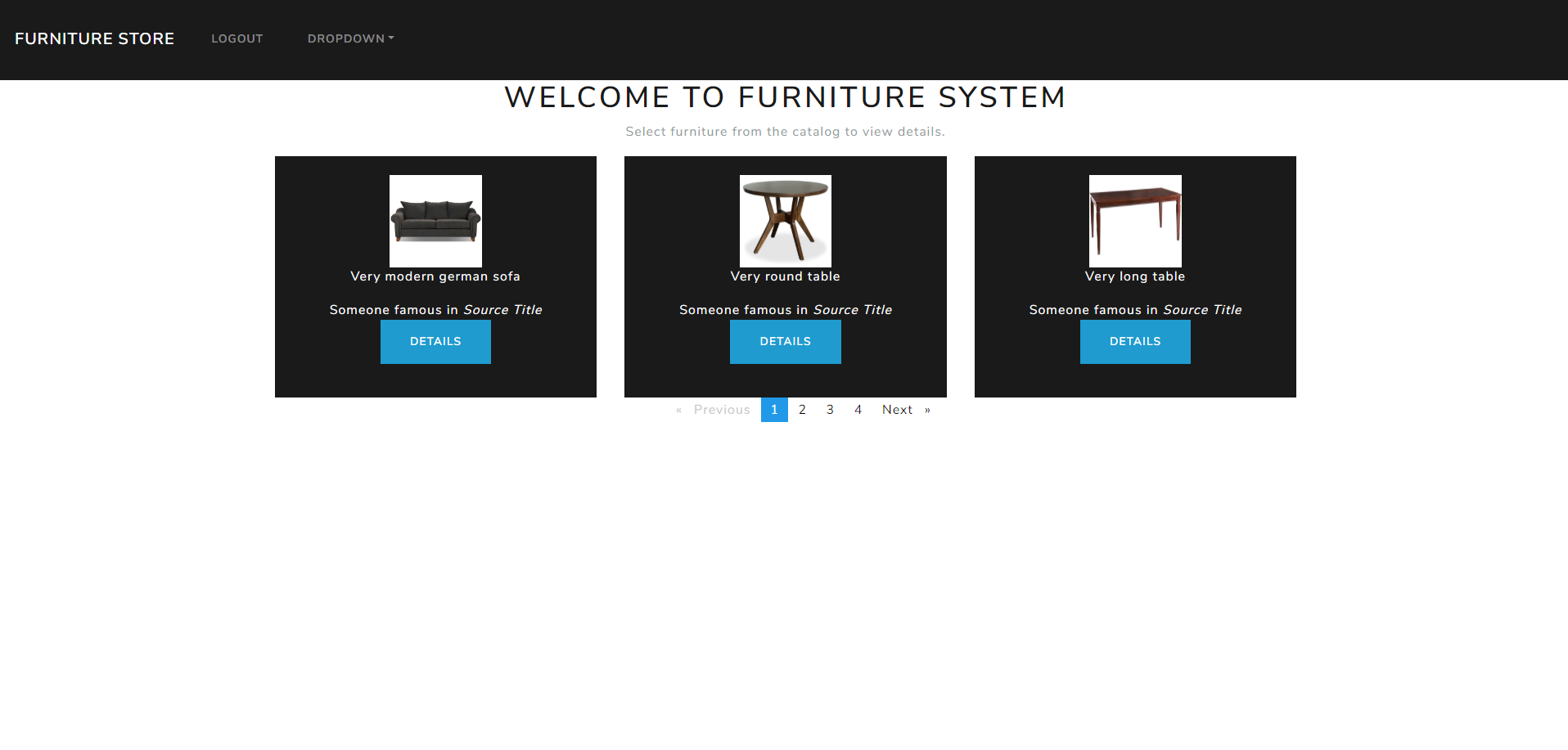
After that we need to define the **pagination controls** at the bottom of our page which structure our page navigation:



The **pageEvent** function changes the **current page** property inside the component.

More on **ngx-pagination** [here](https://github.com/michaelbromley/ngx-pagination).

You can try to implement pagination in **my furniture** component as well.



## Admin Functionality

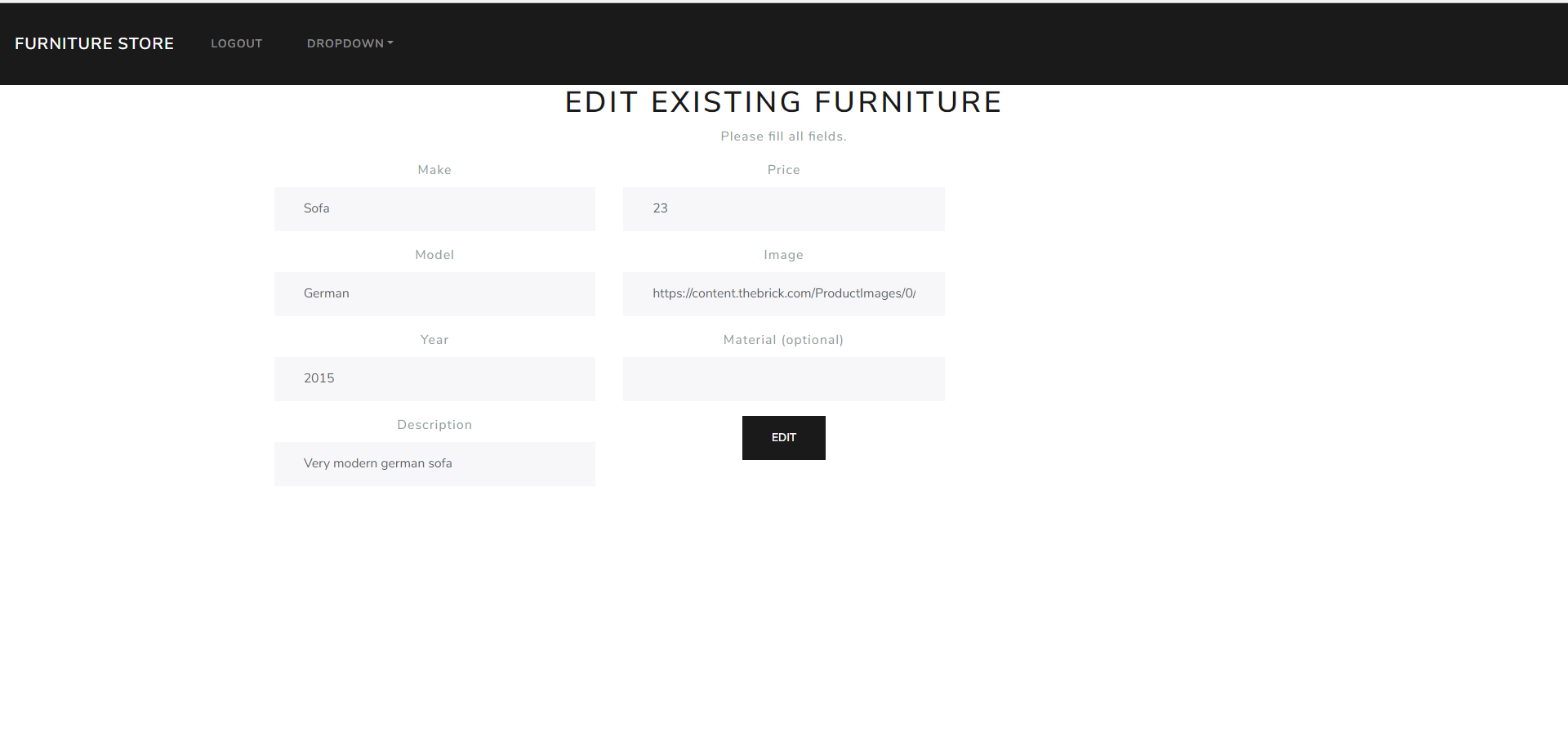
Download the **new server**. It currently supports **admin functionality**. The **first** registered user into the system is automatically an admin (try it out with postman).

Add admin functionality in your angular app by allowing the admin to **delete any** furniture at **furniture/all** and **edit** **any** furniture.

* Furniture by id (GET): **http://localhost:5000/furniture/:id**
* Edit furniture (PUT): **http://localhost:5000/furniture/edit/:id**

Copy the HTML template from **create furniture** component and **modify** it to work for the edit component:

Now we have a fully functioning **CRUD** application ☺



## More on JWT

There is a library that provides Http Interceptor which **automatically** attaches JWT’s to Http Client requests.

[Check it out](https://github.com/auth0/angular2-jwt). It also gives us additional functionality like seeing when a token **expires** or **decoding** a token.