Exercise: Routing

Exercise problem for the "JavaScript Applications" course @ SoftUni.

> **Working with Remote Data**
>
> For the solution of some of the following tasks, you will need to use an up-to-date version of the **local REST service**, provided in the lesson's resources archive. You can read the documentation here.

1. Implement Furniture Store
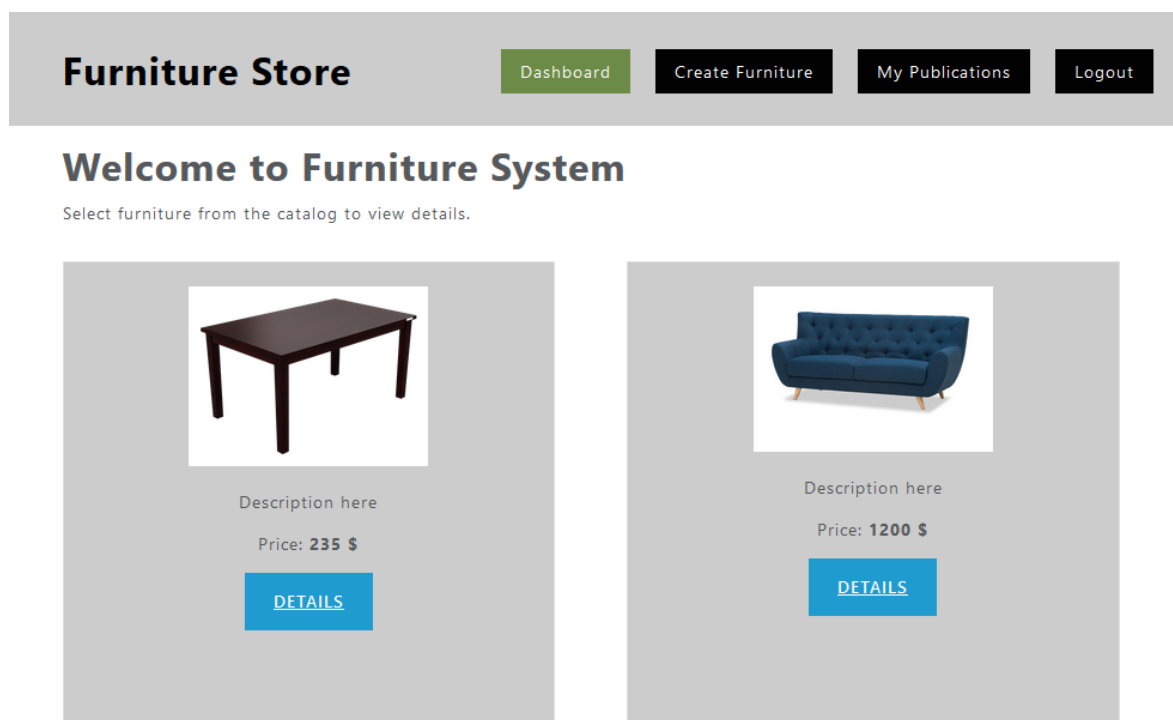
You can make requests to the following endpoints:

- Register User (POST): **http://localhost:3030/users/register**
- Login User (POST): **http://localhost:3030/users/login**
- Logout User (GET): **http://localhost:3030/users/logout**

- Create Furniture (POST): **http://localhost:3030/data/catalog**
- All Furniture (GET): **http://localhost:3030/data/catalog**
- Furniture Details (GET): **http://localhost:3030/data/catalog/:id**
- Update Furniture (PUT): **http://localhost:3030/data/catalog/:id**
- Delete Furniture (DELETE): **http://localhost:3030/data/catalog/:id**
- My Furniture (GET): **http://localhost:3030/data/catalog?where=_ownerId%3D%22{userId}%22**

All Furniture

List **all furniture** inside the **store**. Display appropriate links in the navigation bar, based on user session.

Clicking on any of the Details buttons should **redirect to details**.

Follow us:

Register User

You need to write the functionality for registration of new user. By clicking the "Register" button you have to load the registration form. When the "Register" button of the form is clicked you need to send a post request.

If the registration is **successful** you can **redirect to dashboard page**.

Login User

If the **user has already registration**, the user can login by using **the login form**. After **successful** login the user should be **redirected to dashboard page**. Save the returned token in the session storage, and send it with every request.

Logout User

The **logged in user** can be **logged out** by clicking the **logout button**. Write the functionality for this action.

Create Furniture

Create New Furniture

Please fill all fields.

Make

Price

Model

| Valid input

Image

Year

| Invalid input

Material (optional)

Description

CREATE

Validate fields:

- Make and Model must be **at least 4 symbols long**
- Year must be **between 1950 and 2050**
- Description must be **more than 10 symbols**
- Price must be a **positive number**
- Image **URL is required**
- Material is **optional**

By valid input you can **add the "is-valid" class** to the input field, and **by invalid** the **"is-invalid" class.**

If the creation is **successful** show **redirect to dashboard page**.

Furniture Details

Get the **id** from the **URL** and **display** the information. If **the logged user is the creator** the buttons "**Edit**" and "**Delete**" should be **visible** (can be used).

Furniture Details

Make: **make**

Model: **model**

Year: **year**

Description: **description**

Price: **price**

Material: **material**

EDIT   DELETE

Update Furniture

If the **logged in user** is **the creator** then can **edit** the furniture details. When the form is **loaded all the fields** must be **filled up** with the **information from the server**. **Validation** should be the same as the validation by creating new furniture. **PUT request** must be send.

Edit Furniture

Please fill all fields.

Make

Table

Model

Swedish

Year

2015

Description

Medium table

Price

235

Image

/images/table.png

Material (optional)

Wood

EDIT

Delete Furniture

By clicking on "**Delete**" button the app needs **first confirmation for deleting** (you can use alert or another custom made notification) then sends **DELETE request** to the back-end and **deletes the furniture**. Then the app **redirects** to the **dashboard**.

My Furniture

A logged-in user can see a list of their own publications by clicking the link **"My Publications"** in the navigation bar. Display a page similar to the main catalog (dashboard), but only show the records that are associated with the currently logged-in user.