**Lab: Data and Authentication**

Problems for in-class lab for the "[JavaScript HYPERLINK "https://softuni.bg/courses/js-applications"Applications" course @ HYPERLINK "https://softuni.bg/courses/js-applications"SoftUni](https://softuni.bg/courses/js-applications)

**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](https://github.com/softuni-practice-server/softuni-practice-server).

* **1. Cookbook – Part 2**

Now you will extend the application from previous exercise. Write a JS functionality for register, login and logout. Also logged in users can create new recipies. You are again **provided with skeleton** (**HTML & CSS**) for this task, and you can use the same **server**.

The application now has a navigation bar, with links to other pages. You need to update the visibility of the buttons in the header, depending on whether there is a logged-in user or not. This is how the header should look like for a guest user:



When there is a logged-in user, the header should look like this:



To achieve this, change the visibility of the corresponding **<div>** element.

**Data**

For this version of the app, the data source will be changed since we need to implement user profiles. Use the following endpoints:

* To obtain a list of all recipes:

**GET** [**http://localhost:3030/data/recipes?select=\_id%2Cname%2Cimg**](http://localhost:3030/data/recipes?select=_id%2Cname%2Cimg)

* To obtain details about the selected recipe:

**GET** [**http://localhost:3030/data/recipes/:id**](http://localhost:3030/data/recipes/:id)

* To create a new recipe:

**POST** [**http://localhost:3030/data/recipes**](http://localhost:3030/data/recipes)

Note that for the last request (recipe creation) you will need to provide **user credentials** via the following request header:

**X-Authorization: *{token}***

Read the following sections, to see how the token is obtained for a registered user.

**Register**

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 to this URL: [**http://localhost:3030/users/register**](http://localhost:3030/users/register)



If the registration is successful you can redirect to catalog page. The service will respond with an **authorization token**, that you must save in the browser, using **sessionStorage**. With every following request, send the token in the authorization header.

**Login**

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

The **URL** for login is: [**http://localhost:3030/users/login**](http://localhost:3030/users/login)



**Logout**

The logged in user can be logged out by clicking the logout button. Write the functionality for this action. You can use this URL for logout (send GET request): [**http://localhost:3030/users/logout**](http://localhost:3030/users/logout)

**Create Recipe**

The logged in user can also create new recipes. In the form you have to fill the following fields: **name**, **img**, **ingredients** and **steps**. The ingredients and the steps should be written each on new line, and then converted to an array splitting the text on line-breaks (use the line-break character "/n").



After successful creation of the recipe, you should redirect to the catalog page, where the new recipe should appear.