

Lab: Data and Authentication

Problems for in-class lab 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. Cookbook – Part 2

The resources for this task are available in the following GitHub repository:

<https://github.com/viktorpts/js-apps-workshop>

You may check-out the repository or download the files via the green button labeled "Code" in the upper-right corner. Use the files located in **lesson-03/base** to begin the task. Before starting, make sure you have the most recent version of the repository. To see the solution, check the files inside **lesson-03/finished**.

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 recipes. 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

- To obtain details about the selected recipe:

GET <http://localhost:3030/data/recipes/:id>

- To create a new recipe:

POST <http://localhost:3030/data/recipes>

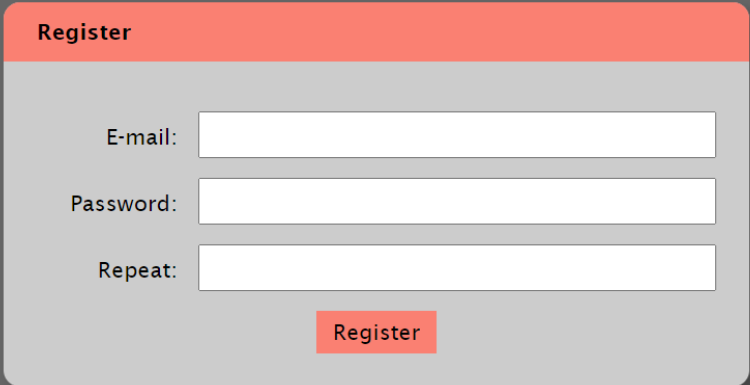
Note that for the last request (recipe creation) you will need to provide **user credentials** via the following request header:

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>

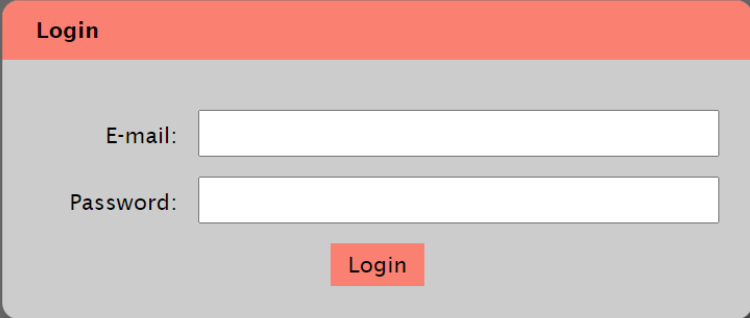
A registration form with a red header labeled "Register". It contains three input fields: "E-mail:", "Password:", and "Repeat:". Below the fields is a red "Register" button.

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>

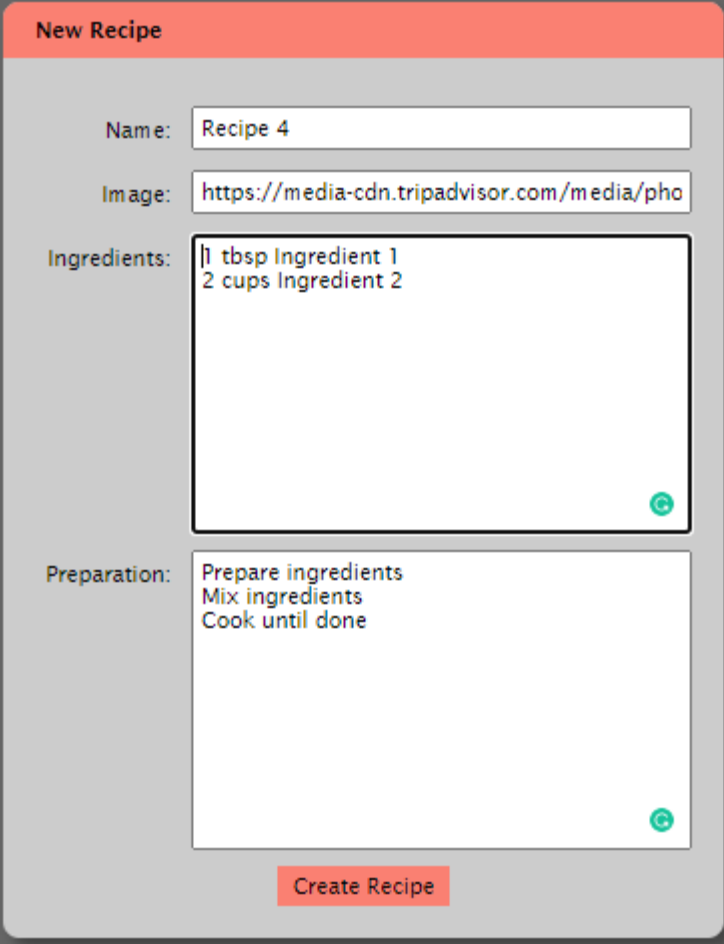
A login form with a red header labeled "Login". It contains two input fields: "E-mail:" and "Password:". Below the fields is a red "Login" button.

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>

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"`).



New Recipe

Name:

Image:

Ingredients:

Preparation:

Create Recipe

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