Lab: Architecture and Testing

Problems for exercises and homework for the "JavaScript Apps" 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. Accordion – Testing

This task is to write tests for the functionality of the Accoridon problem from the previous lessons. You are provided with the working app and need to test the following:

Testing: load titles

First you need to test if all articles are loaded and showed on the webpage with the given titles from the server.

Testing: button functionality

The second thing you need to test is the functionality of the button in the article. Test if by clicking the button the probgram will get the content from the server and show the article body with the right content loaded in it and if the button is named "Less".

Testing: button functionality

The next thing for testing is by clicking the same button again, the body of the article should hide and the button should be named "More".

2. My Cookbook – Testing

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-05/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-05/finished`**.

This task is to write tests for the functionality of the "My Cookbook" app. You are provided with the working app and need to test the following:

Testing: Catalog

- The catalog page should load and render the content of the API
- Displays the recipe details

Testing: Authentication

- "register" makes correct API call
- "login" makes correct API call

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Testing: CRUD operations

- "create" makes correct API call for logged in user
- The author can see the "Edit" and "Delete" button
- "edit" loads the correct article data for logged in user
- "edit" makes correct API call for logged in user
- "delete" makes correct API call for logged in user

3. My Cookbook – Refactoring

Refactor the application, so separation of ceoncerns and other best practices are followed. The following functionality can be abstracted into it's own module, so that the views only contain business logic:

- Requests to the server
- User authentication
- Navigation and view switching
- DOM rendering

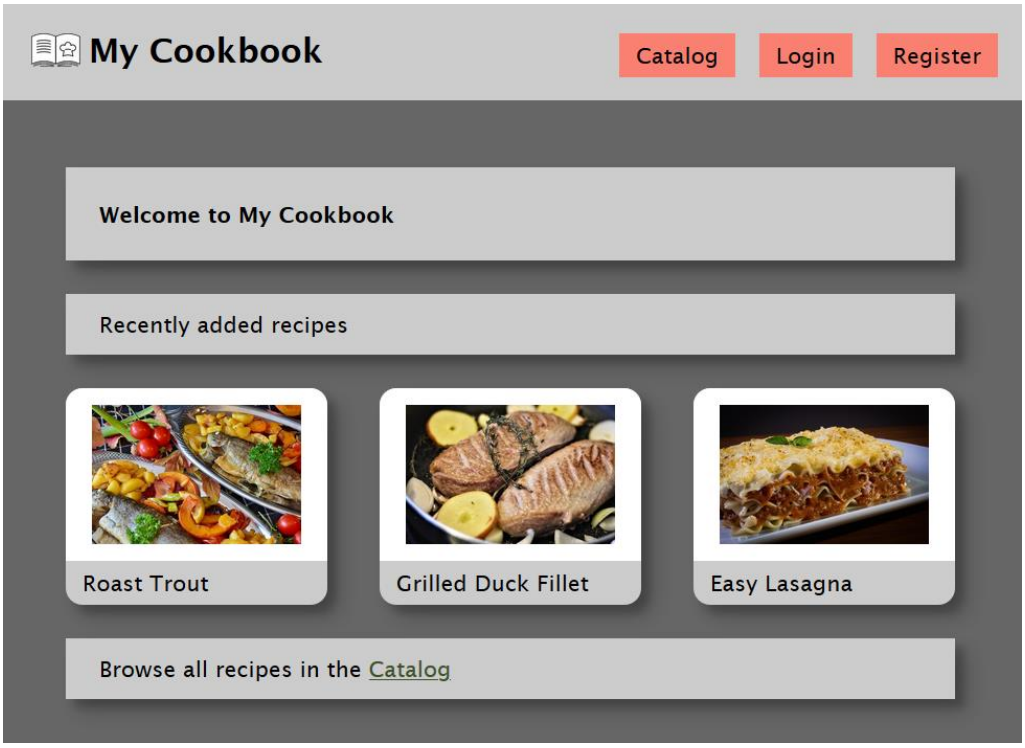
The tests that you created in Task 2 will be very helpful – as you change the code, by running the tests you will always know if everything works as expected or if a bug has been introduced.

4. My Cookbook – Part 4

Home View

Create **automated tests** for this functionality. It's your choice whether to create the tests first (**Test-Driven Development**) or to write them during or after the implementation of the functionality.

The home page contains a welcome message and a preview of the three most recently added recipes, from the newest to the oldest. The application now starts in this view, instead of the Catalog. Clicking on the application title (top left) takes the user back to the home view.



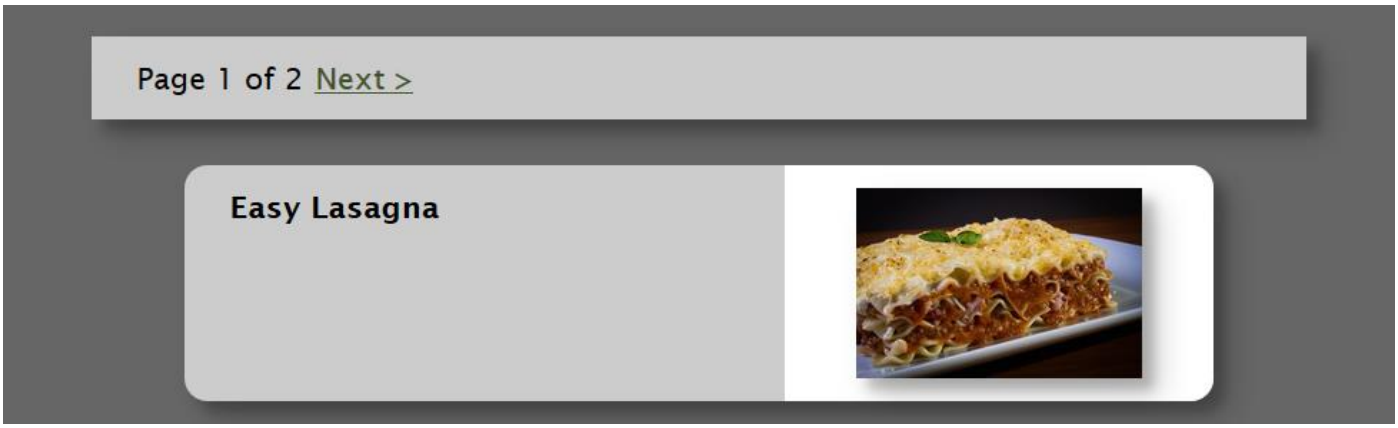
© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

- Get the three most recent recipes:
  `/data/recipes?select=_id%2Cname%2Cimg&sortBy=_createdOn%20desc&pageSize=3` (GET)

Catalog Pagination

Create **automated tests** for this functionality. It's your choice whether to create the tests first (**Test-Driven Development**) or to write them during or after the implementation of the functionality.

Implement pagination for the catalog. Each page must hold 5 recipes. Display page controls at the top and at the bottom of the page.



- Get a page of recipes (**offset** is the number of recipes to skip):
  `/data/recipes?select=_id%2Cname%2Cimg&offset={offset}&pageSize=5` (GET)
- Get the total number of recipes: **`/data/recipes?count`** (GET)

© SoftUni – https://softuni.org. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.