

Knockout exercises

You can find a good reference to JS here <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide> , to jQuery here <http://api.jquery.com/>, and to knockout here <http://knockoutjs.com/documentation/introduction.html>.

Exercise 4.2.1

Use knockout.js and the knowledge you have from sec 2 to solve this exercise. This exercise should be based on on the VS project: https://github.com/bulskov/RAWDATA2018F_Exercise4_2_StarterProject which contains a simple backend that returns posts. You should add to this a frontend that implements the requirements described below. In this starter project all the JS/CSS libraries you need is already included (require, require-text, jquery, knockout, bootstrap).

The solution must use knockout.js for data binding and require.js to handle module loading. The following requirements is a stepwise extension of the web page. Use the revealing module pattern for the viewmodel(s). Feel free to add any CSS styling to the page to make the layout nicer 😊

Note: knockout.js has very good documentation (see footnote)

- a. Create a page that can show posts. The data should be fetched from the backend using the “api/posts” path.
- b. Extend the list (a) with two buttons, “Prev” and “Next” such that the page can switch between pages. If there is no previous or next pages the respective button must be disabled.
- c. Make the posts in the list implemented in (a) links bound to the URL given by the backend.
- d. Wrap the parts in a,b,c into a knockout named template, by use of the SCRIPT tag with type “text/html” and an id, and introduce a new DIV tag that bind to this template.¹
- e. Create another named template with html for showing a specific post.
- f. Make the binding in the DIV (d) dynamic, such that changing the name of the template will change the content of the DIV, and thus the page itself.
- g. Bind click on the titles in the list (c) to a function that can change template, fetch the post from the backend, and show post details.
- h. Add to the details in (g) also all answers.
- i. Add a back button/link on the details page to get back to the list of posts.
- j. Remember the page and page size so you get back to the page you left from.

This exercise will be extended in the exercises next week with knockout components, and will then be a small, but real, example on how to do the basic html/knockout/ajax mechanics for your portfolio project.

¹ <http://knockoutjs.com/documentation/template-binding.html>

Exercise 4.2.2

Github Viewer

Use the web service provided by Github² to create a viewer with the following requirements:

- When the user enters a valid userid a list of the repositories for that user should be listed with name, stars and language.
- If the input is not a specific user then a search for users should be shown instead – using the input as the query, and each of the results in this result should be clickable to get to a).

Github Viewer

Name	Stars	Language
knockout	7412	JavaScript
tko	6	JavaScript
tko-policy	2	JavaScript
tko.bind	0	JavaScript
tko.components	0	JavaScript
tko.computed	0	JavaScript
tko.observable	0	JavaScript
tko.utils	4	JavaScript

² <https://developer.github.com/v3/>