

Projektarbeit

“Shopping List”

Aufgabenstellung

Wählen Sie ein JavaScript Frontend-Framework (Angular / React) aus und programmieren Sie eine Einkaufsliste. Die Web-Applikation soll die grundsätzlichen CRUD Funktionalitäten erfüllen. Eine Auflistung der Anforderungen finden Sie unten.

API Model

Ein Eintrag in der Einkaufsliste ist in der API folgendermaßen definiert (siehe „angabe-db.json“ als Vorlage):

ShoppingListItem

- | | | |
|-----------------|---------|---------------------------------------|
| • id: | number | [auto-generated by API / JSON Server] |
| • title: | string | „Milch“, „Eier“, „Mehl“ |
| • description?: | string | „Mehl glatt“ |
| • amount: | number | 1, 500, 0.5 |
| • unit?: | string | „Stk“, „g“, „liter“ |
| • done: | boolean | true false |

Anforderungen (Requirements):

Folgende Anforderungen und Features soll die Applikation abdecken. Diese sind als UserStories formuliert.

- **As a user, I want to view all ShoppingList items.**
Make sure that it is clearly visible if an item is done/not done. Maybe use different colors for done / not done items.
- **As a user, I want to toggle an item between done/not done.**
I want a convenient way to mark an item as done or not done. That could be a button or double clicking an item in the list. Make sure that it is clear to the user how to toggle especially if no button is used.
- **As a user, I want to delete an item from the ShoppingList.**
I want to delete items from the ShoppingList. It is okay if the item is deleted without a confirmation dialog, although it does not hurt either (nice to have).
- **As a user, I want to add a new item to the ShoppingList.**
I want to add a new item to the ShoppingList using a Form on the page. The Form should consist of the following input types:

title:	text
description:	textarea
amount:	number (min = 0)
unit:	text (or a selection list with predefined options [“Stk”, “Liter”,])
done:	checkbox

- **Styling (CSS)**

Es steht Ihnen frei Frameworks (Bootstrap) o.ä. zu verwenden. Die Anwendung sollte zumindest über einen ansprechenden Header und eine übersichtliche Ansicht der ShoppingList Elemente verfügen.

Nicht-funktionale Anforderungen

- Verwenden Sie Angular oder React.
- Als API-Mock verwenden Sie bitte JSON-Server [1].
 - Als Vorlage für ihr db.json verwenden sie bitte die beigefügte angabe-db.json.
 - Stellen Sie Testdaten in der Abgabe bereit. D.h. die Abgabe soll eine db.json Datei mit Daten enthalten. Die Applikation soll beim Starten nicht leer sein.
- Operationen gegen die API müssen asynchron ablaufen.
 - Vor allem für den API Service wichtig, aber nicht ausschließlich.
- Kommentieren und Formatieren Sie den Code für die Abgabe.
 - Denken Sie daran: Beschreiben sie „Warum etwas so gelöst wurde.“ und NICHT „Was jede Zeile tut.“. Zielgruppe für Kommentare sind andere Entwickler/innen und die können Code lesen 😊.
 - Lesbarer Code ist Kommentaren immer vorzuziehen!
- Der „node_modules“ und „.angular“ darf NICHT abgegeben werden!!! Das würde das zip-file auf eine Größe anwachsen lassen, die vom Mail-Server geblockt wird.
- Bitte komprimiert (zip) eure Lösung und benennt diese „{Nachname}_ShoppingList“ und schickt diese an stefan.erlachner@bitschulungszentrum.at mit dem Betreff „[Projektarbeit] ShoppingList“.

Bonus Anforderungen (nicht verpflichtend)

- As a user, I want to filter the ShoppingList items if they are done / not done.
Filter options:
 - All
 - Done
 - Not done
- As a user, I want to sort the ShoppingList items.
Sort options:
 - By Id
 - By Title
 - By Unit

Links:

[1] <https://github.com/typicode/json-server>