INF-2300

Assignment 2, To Do List Web Application

Magnus Dahl-Hansen

October 2022

Introduction

This report describes the implementation of a web application that gives the user of the program the possibility to handle a To Do list. The functionalities of the list are to add items to a list, delete items, mark each item as “completed” and remove all items from the list.

Technical background

Before reading how the code was implemented, the reader of this report should be familiar with: React and HTTP and HTTP requests,

React is a free JavaScript library for making user interfaces and single page applications. In the React library one can use special functions called “useEffect” and “useState” which enables one to perform some side effect or modifications on a component or to use a variable in a special state. The *effect* function is called only when a variable associated with it is updated or modified. The *State* function on the other hand enables one to pass a variable to a function in its current state to for example gather some input from the user.

HTTP stands for Hypertext Transfer Protocol and is the most common protocol to fetch resources and load web pages over the internet. By sending HTTP-requests different internet communication platforms are able to ask for resources and information that they need to load a website. The different HTTP-request used in this implementation are: “*Get*” which fetches information from a server, “*Post*” which places information on the server, “*Put” which* replaces information on the server and “*Delete”* that deletes information from the server. In addition to the React library the *Axios* library is used to handle these different types of HTTP requests.

Design&implementation

In this implementation all functions, variables and the rendering process of the screen is set up in one function called “Main”. All headers, buttons and input bars are displayed in the browser by a return statement at the end of the *Main* function which is made possible by the React library.

In the *Main* function the variables and the *state&effect* functions are enabled. There is created one *effect* function for each of the *Get*, *Post*, *Put* and *Delete* functions and one *state* function for the input bar which are called when their respective button are pressed. Out of the mentioned functions only the *Get* effect will run when the application is started to retrieve the items located on the server and display them. The items which are fetched are three chores which are directly added to a list called “*To* Do” and displayed on the website.

Upon running the application an “Online To Do List” header is displayed along with an input bar and a add-button. When the user inputs a string to the input bar the value of this string is fetched using the React-*state* function and saved inside a local variable. If the user chooses to press the add-button the variable holding the input string is added to the *To-Do* list and loaded to the server using Axios’s *Post* function. After updating the list to the server the variable associated with the *Get* effect is updated which in turn will call the *Get* effect and display the newly updated list to the web browser.

The functions that modify and item or deletes one or all items from the *To Do* list are similar in many ways. They all have in common that they are called when the variable associated with each of them is modified, they all use the *Axios* library functions to perform either a *Get*, *Put*, *Post* or *Delete* request and they will all in the end reset their respective variable.

The main difference between the functions is that the *Put* and *Delete* effects must be called with a specific item id so that they can modify or delete the item from the *To Do* list with the matching id. How to find the specific item based on the id is handled by the URL that is passed to the Axios functions. The effect that deletes all items from the *To Do* list will perform a Axios delete function on all elements in the *To Do* list through a loop.

Results&discussion

The implemented code is fully able to render a user interface that allows a user to add and delete items, mark an item as completed and clear the whole To Do list. The interface will on startup load and display three elements which are already located on the server.

However there is one issue that has not been solved which occurs upon reloading the web page. The issue is that the checkboxes that marks a chore as completed will disappear if the web page is refreshed.. This issue appears in the *Put* effect and may be caused by the “axios.put()” function. Upon trying to modify the “name” of an item the “axios.put” function works as expected and will modify the correct item, however when trying to modify the “done” variable of an item it will not work correctly.

Conclusion:

This assignments task was to create a web application that allows a user to create and handle an online To Do list. The implementation of the functionalities: add, delete, mark chores as completed and getting items from the server is described. The web application is able to run successfully with one minor issue that the checkboxes will reset upon reloading the page.