Project in full stack development

Lab 1

Simon Eriksson

TGDU3

1. Tasks and implementations  
   I have been tasked with creating a basic REST API. After creating the API, I also had to add CRUD operations to it. The goal is to manage a music albums collection through a web-application, which is connected to a cloud database where all the album data is stored.  
     
   I initially started by mapping out the routes and creating a basic layout for them. I won’t go into much detail here since most of it is just boilerplate routing code. Creating the cloud database and connecting it to my application was also straightforward.

I found mongoose to be very convenient and easy to use, making the process of connecting the app to the database a non-issue.

For a smaller project such as this, I did not put much effort into organizing files, which I would have done had it been a bigger project.

All the static HTML contents such as *‘find album by name’* and *‘add album’* were hard-coded since rendering it on page-load through JS seemed pointless.

I did however render all the “dynamic” content using JS. This would include generating cards for each album, and the required buttons for managing the data entries.

I wanted to be as consistent as possible by associating each button/operation to the ID of the data entry. Meaning all the buttons and features of a HTML-card would be identifiable though the database ID associated with its graphical representation. I’m quite happy with this implementation, even though improvements can be made.

En bild som visar text

Automatiskt genererad beskrivningEn bild som visar text

Automatiskt genererad beskrivning

The screenshots above are a good example of my thought process behind some of these implementations. I wanted to use the ID for identifying which object I wanted to modify. Since the event listeners are part of the HTML-card rendering script, each rendered card should be associated with the proper ID in the database. Meaning each event listener is attached to the ID of the frontend card contents.

This is also the main reason why I decided against manually adding an ID-field for the album creation form. The automatically generated IDs require less maintenance and would let me manipulate the data without having to deal with any inconsistencies.

1. Issues

I faced some issues where some query selectors would look for DOM content that had not been rendered yet.

Wrapping the code in: document.addEventListener(‘DOMContentLoaded’) seems to have solved that issue.

1. What I’ve learned

This required quite a bit of work on the HTTP-request side. Learning how to set up more complicated requests along with HTML rendering was a good learning experience. I spent most of my time working on the event listeners, making sure they were sending the correct values.