ARCADA

Personal report - Cloud Native Apps (2022-23)

CNA - Order service

Karim Kanji
IA-20
2022

Table of contents

Github information

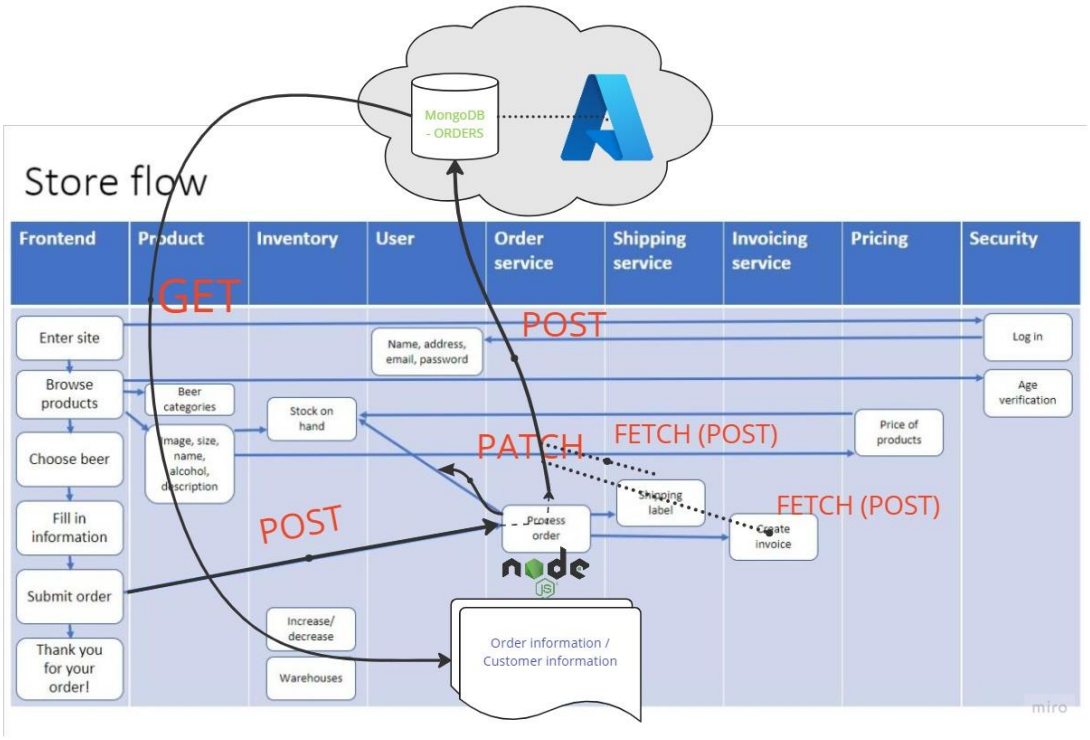
Link to the GitHub repository: Repo
Direct link to the API spec for order service: API Spec

Introduction

In this course I worked in the order service with Sebastian Fallström and Oscar Weber. The order services primary focus was to receive the orders that come from the front end and serve as a hub for the entire business. There order process is one of the more critical parts of the entire business considering that the customer won't receive their items if we don't do our part, nor will the business receive money. To fulfill our task as the order service, we used NodeJS as our programming framework, MongoDb as our database (+Mongoose) and Azure to host our service (Orders (Takes a while to load ~2min)). To access our API you need to use the JWT token provided by Security (Ville Hjelt)

Flow chart

To further explain how the order service interacted with the other services, I made a flow chart which shows the requests we make to different services.



To explain the flowchart, we gave our POST link to front end which they in turn apply to the form that the user fills in when they want to purchase something from the store. Our POST request does 3 things:

1) Sends the order information to our database
2) Does a Fetch (POST) request to shipping service
    a. This is done so that shipping service can also get the same information we get. This way they can process the information with their own methods in order to fulfill their part in the business. For example, to produce a shipping label with the customer information.
3) Does a Fetch (POST) request to invoice service
    a. Self-explanatory, the invoice team needs all the information from the order to accurately charge the customer.

> Side note, both the invoice team and the shipping team need to have their own databases so that they can manage the status of their respective task for example “Invoice/Shipping done: true / false”

We also have a GET request which we can use to get a list of all the orders and unique customer information (unique in this context meaning when we get the customer information there are no duplicate customers, which is important of course since a single person can make multiple orders).

Learnings

It was obvious that this course did not focus on difficult programming, rather on the workflow and cooperation between the different micro services. I have learned a lot about communication. It is extremely important to early on make a MVP so that other services who are dependent on you have a place to start. If proper communication between groups isn’t prevalent it leads to a spiral of problems and incompatibilities between the different services. I also learned how to better structure priorities in these kind of projects. The most important thing is to quickly get some sort of prototype up and running so you can get a better understanding of what is going to work in practice and what isn’t and communicate that with other groups. This course put my communication skills to the test. I learned to think about not only our service but also other people’s services when coding and making decisions, which is something we haven’t had to think about yet in IA-20.
We were also supposed to have a PATCH up and running with inventory but due to their lack of communication, structure of their service and slow progress, we really struggled with getting that up and running, by the time they had things working we were already done with the course (as they mentioned during the presentations). A similar case occurred with the invoice service. They got their endpoint up and running a couple of hours before the final presentations which caused us to quite rapidly need to get our services to work together. We had however prepared for this situation and we hade made working inactive code that only needed the link to their API to fully work. Therefore, we were able to quickly get our services working together before the presentation. Another learning from this course is to not procrastinate since other people and services are dependent on your own.
There were not too many things code-wise I learned from the course, however there were a few new concepts: Fetch and CORS. These concepts were entirely new to me, and I feel like I have a much better grasp of these now than I did previously

Contribution

My contribution was mostly to code, organize and talk about our groups progress on the status checks. I also discussed with other groups a lot about our vision of the data flow and made sure that other groups were up to date. Besides the above I also established our database on MongoDb and presented our group work on the final lecture. I would say the bulk of my contribution was the communication and organizing both within our group as well as with other groups. Quite many nights were also spent on trying to solve bugs or other problems with the rest of the group.

Other team members contribution

I have worked with my group members on many projects before. We know each other’s strengths and weaknesses and were therefore able to accurately split the work so everyone had comfortable and appropriate workloads. The entire group did lots of work to get our service up and running and working correctly. Sebastian and Oscar spent more time on the code than I did, but in return I handled more of the leading, organizing, and communication. All in all, I think this was a very well executed project and the entire team equally contributed to the team’s deliverables.