

Lappeenrannan teknillinen yliopisto

Software Development Skills, Fullstack

Janni Timoskainen

LEARNING DIARY

17.09.2023 NodeJs

I started going through the courses Moodle-page and got a brief thought that what this course contains. I created the git repository for this course. I watched the first video.

NodeJs is JavaScript runtime so it isn't a language or framework. Node is fast (it runs on a single loop), efficient and highly scalable. NPM is node package manager which allows to install 3rd party packages. Node has core modules like path, fs, http etc.

I coded at the same time with the tutorial, so I learnt a few things. For example,

- these `` are used in JavaScript so that one can use \${ } inside of a string.
- I also learnt different ways to show the paths.
- Nodemon is excellent to get the page to refresh without manually (^c) executing the run in the terminal. Also other server related stuff was a bit unclear to me before.

I've done courses like front-end and started the Introduction to web programming so the css and html part was already familiar for me.

18.09.2023 MongoDB

I watched the second video and coded along with it. Mongo was totally new thing for me so basically everything was new. I know postgresql so I have a tiny clue what is coming.

MongoDB is noSql-database, which stands for not only sql. Collections of documents. So, basically it will work as the database for the application in the making.

ExpressJS

Not high-level framework, minimalist web framework for Node.js. It's back-end framework. This makes the web applications with Node.js much easier.

Middleware functions are functions that have access to the request and response object.

In the very beginning of the express crash course, I realised that this is way easier than in the node crash course. This is mostly because the static folder and of course express.

Middleware functions have access to request and response. I learnt how APIs are constructed and learnt to use the postman.

19.09.2023 Finished the ExpressJS module

I think it's really interesting to see how an app is constructed via the backend as well as the frontend. I have learnt a lot of new things but at least now they've felt alright.

26.9.2023 Starting the Angular module

I started going through the angular module. I decided to start with the building of my first Angular app. That was actually just downloading the Angular so next I started the "Tour of Heroes" tutorial.

I found this video [Let's Learn Angular Together! - Setting it All Up - Angular Tour of Heroes Tutorial Part 1](#)



That basically teaches the tour of heroes, so I coded along with it. I didn't find any video from the moodle or from the documentation so I searched from Youtube to help me visualize the tasks a bit easier.

Pretty much everything in this one is new for me. So, I learnt how to create the base for the app. I also understand the connections of the different files now better.

27.9.2023 Finishing Angular module

I started the third part of this tutorial, "Create feature component". I learnt how to make two components that are connected together. This part was quite easy, just separating the two components from each other.

I finished all of the parts. I had a bit of a struggle with the service stuff, but I managed to do this part also. Creating the python api was interesting. The last part was quite heavy with the modifications but I managed to go through them.

6.10.2023 MEAN-Stack

I started doing the project of this course with MEAN stack. I followed with the tutorial and also coded the project itself. I created new repository for it. For me this was the module I learnt most of as I got to merge all the subdivisions together.

Lots of the functions used in the tutorial used callbacks that aren't anymore supported which caused a lot of problems, but I managed to get through them. Basically, the functions needed to be changed to async functions and promises.

8.10.2023 Continued the project

I fixed the authorization as I had some problems with the async functions. I couldn't figure out a way to use the flash service and that way the flash messages, so I created somewhat similar notices via html. I didn't want to change the version of the Angular to older one (would have had to change the node too) so I thought a different way. I think that works ok. The angular2-jwt was also some old stuff and I decided to check if the user is logged in via the local storage as in the "loadToken()" function. This worked as intended.

11.10.2023 Finalizing the project

I decided to finish the project via simply adding some elements I wanted to have. I also think that I will continue this project on my freetime to make it better. I wasn't able to finish the project yet, as I couldn't get the message bar working.

2.11.2023 Finalizing the project

I didn't implement the message field as I couldn't get it to work. So now the final project contains the page in which a user can register. The registration information is saved then to the MongoDB. The user may see the dashboard when signing in and can log out.