

Lappeenrannan teknillinen yliopisto
School of Business and Management

Software Development Skills

Learning Diary Fullstack Module

Lauri Tirkkonen, 0613034

Software Development Skills: Full-Stack

5.5.2023

Having completed the advanced web applications course, I felt confident in the technologies used in the course: Node, MongoDB and Express. Angular was new for me, but I had heard of it before. I figure it will be used for the frontend similar to React. Even though it was recommended that you should push to github often, I chose not to as I don't know how to push folders there and all together seems confusing essentially pushing 4 different projects to a single repository, but I'll figure it out at the end.

NodeJS

As I still have lot of what was done in the Advanced Web Applications in memory, the theory / info in the beginning was pretty much recap.

I feel like I possessed the coding fundamentals, but the go through of the modules was interesting and clearly showed the abilities of these different modules. I chose not to install the nodemon in the dependencies as I have it installed globally (not sure if it was mentioned this should not be done). I neither deployed the project in heroku as it seemed simple and clear, so I felt I wouldn't gain anything from doing it personally.

MongoDB

At first glance the Mongo portion seems as well that it will mostly be repetition. And so, it mostly was. I am not either sure should the queries shown in the video be written into text file or something and returned, but it seems somewhat stupid, so I didn't do it. I have used Atlas only once, so that was as well interesting, but I am familiar with the MongoDB Compass.

6.5.2023

ExpressJS

I was familiar with Express before, so again this was mostly repetition. The use of handlebars as a view engine was new. I always find it so annoying as you have to be so careful with the different versions. I initially installed version 7, while the tutorial was using 3.

7.5.2023

Angular

I was not familiar with Angular, even though I had heard of it, but never used it. Angular did not work with node version 12, and node version 14 -> was required so I just used the latest version I had v19.5.0. Once again, I hate the problems relating to the versions and how these examples have been made years ago and the older versions are no longer supported etc. But I got Angular working.

I'm comparing Angular to React as for me it is the closest in similarity and Angular seems more complex. Everything was new, but same principles of course. I learned about Angular framework fundamentals, Angular CLI, Components and services and about events, https and routing in Angular.

After completing the tutorial everything else worked normally except details component didn't create nor include css file, so the details page was not visually correct, instead a spec file was created. I didn't figure this out and still don't know what happened. I tried creating a css file manually, but that obviously didn't work.

MEAN-Stack

I feel like I will use node version 19.5.0 for the Mean-stack example project as I rather fix the syntax problems than struggle with the right versions and problems arising from that. I was once again confident in my abilities as I completed the advanced web applications course, where a similar project was conducted using MERN stack. I didn't know you could write directly to the package.json and then just run npm install. Otherwise, the packages were familiar, and I have used all except passport, but that I as well know.

8.5.2023

The angular/http package has been removed and replaced by angular/common/http. Connecting to MongoDB doesn't work with localhost and instead 127.0.0.1 has to be used. map needs to be used with pipeable operators, no deep imports. In tsconfig.json file I set "strictPropertyInitialization": false to get rid of compilation errors and the need to initialize all my variables. angular2-jwt can't be used anymore with newer stuff, so I couldn't get the application to compile, so I had to use the older versions. The project still complained of "Mongoose: mpromise (mongoose's default promise library) is deprecated, plug in your own promise library instead", but after adding "mongoose.Promise = Promise" everything worked. I didn't go on again with the Heroku as it seems plain and simple, in addition I wouldn't deploy it on Heroku as of 2022 it no longer offers its services for free.