Lappeenrannan teknillinen yliopisto

School of Engineering Science

Sofware Development Skills

Vili Huusko, 0544718

LEARNING DIARY, FULL-STACK MODULE

**LEARNING DIARY**

20.12.2021

I read all the course information on the “Course overview” and the “General course information” tabs. I learned the goals of the course and the required work to do in order to pass it. I chose the full-stack module since it was the most interesting out of all the modules, and I also thought that I had a “good flow” after just finishing the “web applications” course.

Full-stack developing is already a familiar thing for me. The “web applications” course, and especially the project of it, taught me a lot about the subject and sparked an interest towards web development in me. I noticed, however, that this course utilizes the Angular.js front-end framework instead of the React.js framework which I have prior knowledge of.

I was already familiar with Git and I had no problems setting up a repository for the course. Choosing an editor was a no-brainer for me, since I have used Visual Studio for almost half a decade, so that’s what I decided to go with.

I started doing the Node.js module immediately. I am very familiar with Node.js beforehand but I started watching the video knowing that I’ll learn something new since Traversy media videos are very in-depth.

I watched the first 15 minutes of the video. The part that I watched was mostly knowledge I already had. It was, however, a good refreshment.

21.12.2021

I continued to watch the video about Node.js and followed the code example. I learned about the module wrapper functions in Node.js. (I have used them before, never really thought about the inner workings of them). The presented core modules “path” and “fs” were already familiar for me, so I didn’t bother copying the example codes since those wouldn’t teach me anything new. The “os” core module wasn’t familiar to me, so I followed the steps in the video to recreate the file.

I Also did the “url” core module section, even though I’ve worked with that module before, but it was so long time ago that I needed a refreshment. I didn’t remember exactly how to add search params to a url, but I relearned that. I also relearned how to make a simple Logger with event module.

After presenting the other core modules, the video focused on creating a http server with pure Node.js using the “http” core module. This is something I have never done before; I have always used Express.js. I followed the coding example on the video and learned how different it is to create a simple web server with pure Node.js and also just how much Express.js helps with back-end development.

While creating the server, I copied the html and css code straight from the author of the video, since that part wasn’t the focus of this section of the course. While creating the server, I just further learned just how tedious it is to create the simplest server with bare Node.js without any frameworks (such as Express.js).

The last part of the module was deploying the server online in Hekoru. This is something I have never done before and something I’ve meant to look into. After following along how to deploy to Heroku I was done with the module.

I continued with the second module (MongoDB). I was already very familiar with MongoDB, and I’ve used it independently and with the Mongoose library and even developed software with them.

I had already installed MongoDB on my machine, and I already knew everything from the beginning of the video. However, I’ve never used the Mongo CLI, instead I’ve always used the MongoDB Compass GUI software. For the sake of learning I did all the queries with the CLI.

First, I learned how to start the MongoDB shell on Windows. It is done by navigating to the MongoDB/server/bin folder and typing mongo into the terminal. Then, I learned the most basic navigation commands “show” and “use”. After that, I learned how to create and drop databases and collections, how to insert data into a collection, and how to show data in a collection (both formatted and non-formatted). Adding clauses and parameters to finding documents in collections doesn’t really differ from doing it through code, so that was old information to me. Same thing with updating and deleting data.

The main takeaway from this part of the video was the fact that I didn’t see any benefits of using the CLI over the MongoDB Compass GUI program.

Lastly, the video covered the MongoDB Atlas. It is basically a cloud implementation of MongoDB. The topic was familiar to me since I’ve used it before. However, it has been some time since I last used the MongoDB Atlas, so the video was a nice reminder.

I continued to the next module (Express.js). This module is already familiar to me since I have actually completed it previously on my free time. I have also used Express.js extensively lately, so I doubt that I’ll learn much during this module. I still think, however, that I might pick up a few things and get a refreshment on some things that I have not used in a while, such as view engines.

The module started with a set of introductory slides and installation instructions. These contained only old information for me, but the introductory slides had some good refreshment on express middleware. The start of the video also introduced a program called “Postman” which I already have installed and am familiar with.

Next, the video covered setting up an Express server which I also was very familiar with, and that’s where I decided to end the day.

22.12.2021

I continued with the module. I watched the video and decided to change the example program a bit. In the video, the data handling was done with a file writer, I implemented it using vanilla MongoDB without any drivers, since I have never done that before in an express environment. It took me much longer than anticipated to get it to work, but I enjoyed the learning process. I think that it was better to do the module my way, since it supported my learning better.

After implementing the CRUD routes with MongoDB, I started to implement the view engine. The view engine initializing came really fast into my mind and the only problem I had with it was that the method in the video was a bit outdated, so I checked the “express-handlebars” npm site for usage instructions.

In this module I learned way more about MongoDB than in the previous module. I’m glad I saw the extra effort to modify the module a bit and implement it here.

27.12.2021

After the holidays I continued with the Angular module. For me, this was the most important part of the course since I’ve never worked with Angular before. I am, however, not completely unaware of how to work with front-end frameworks since I’ve worked with ReactJS before.

I started off by learning Typescript first. I’ve never used it before, but I have seen some of it in some YouTube tutorials. After further inspection, I decided to spend the rest of the year not learning Typescript and I decided to end the day there.

03.01.2022

I continued to learn TypeScript. I started off by reading the “TypeScript for JS Programmers” and “TypeScript for OOP Programmers” sections of the TypeScript handbook found on the TypeScript official page. After reading said sections, I think I got the gist of the features of TypeScript.

After that, I continued with the Angular tutorial and read the “Introduction” and “Create a project” sections of the tutorial. The introductory section explained what I’ll learn during this course and how the final project will look like.

The “Create a project” section taught me how to setup an Angular environment and how to use the Angular CLI. The workflow of creating an Angular app differs a bit from a normal npm workflow. Instead of doing all the work with npm, Angular utilizes its own CLI that is installed with npm. I had to install the newest LTS version of node in order to get the CLI to work. I finished the “Create a Project” part of the tutorial and familiarized myself with the automatically generated file structure of Angular.

After creating the project and finishing the “Create a project” part of the tutorial I ended the day there.

04.01.2022

I continued working on the Angular module. I refreshed my memory on how to set up the project and use the CLI. After that was done, I continued with the module. After the component was generated, I noticed that Angular has a lot of boilerplate code and lots of files. I have yet to decide whether I like it or not.

Angular uses classes as components. So does React, but React also supports function components, which I have only used. I took some time to read about JavaScript classes to see if there’s anything I should learn about them. After reading about JS classes, I continued the tutorial. I learned about piping, which I found to be a very helpful way to format strings, for example. I also learned about the [(ngModel)]: Angular’s two-way data-binding syntax. I have done similar thing in React using useState hooks, so it wasn’t completely unfamiliar to me. I actually prefer Angular’s way of doing it since the syntax is way simpler and more effortless to write. However, I found the importing to be a bit weird.

I continued to the second part of the tutorial: displaying a list. In this part, I learned about the \*ngFor directive. Again, I’ve done something similar in React using the classic vanilla map(), function. The Angular way of creating lists is more effortless to write but React’s way of mapping the values is way more intuitive for me at least. I also learned about the \*ngIf directive, which I found useful. Also, I learned about Angular’s class binding, which was a bit awkward to write at start, but which I’ll eventually get used to.

I continued to the third section of the module: feature component. In this section I learned about the @Input decorator and property binding. I much prefer React’s way of using props to send data to components.

In the fourth section, I was introduced to services and Angular dependency injection. After creating the service and getting it to work synchronously, I learned about Angular’s way of asynchronous data fetching using Observable class and subscription. The rest of the section covered pretty much the same that has already been covered, however, the Message service was something new. (Injecting service to a service). After completing the part 4 of the tutorial, I decided to end the day there.

05.01.2022

I began the day with the fifth section of the tutorial, which focused on routing. I have implemented routing before in vanilla JS as well as in React (using react-router-dom package), so the concept isn’t totally new to me.

In Angular, the routing is done very differently. It started by creating a routing module with the CLI. However, I had already done this upon the project initialization. The tutorial clearly explained the generated file. I’ve always found routing to be cumbersome in React. However, Angular’s way of routing seems clear.

The tutorial then focused on creating appropriate views for the new dashboard route and moving the “heroes” view to its own route. After that I learned how to do parametrized routing and how to extract id from URL in Angular.

09.01.2022

I decided to at least finish the Angular module today, since there was only the last part of the tutorial left. I started to work on the last part of the tutorial. It focused on retrieving data from a server. I have done this in both React and in vanilla JavaScript using async fetch(), so I was already familiar with it.

I learned about Angular’s in-memory web API used to make demos. I found it weird to use it, since I expected to use actual data fetching from a remote server.

Next, I learned how to fetch data in Angular. I didn’t like Angular’s way of fetching data at all. It felt overly complicated compared to vanilla JS fetch() -method and it required too many steps and difficult syntax.

After the logic for getting a hero was done, the tutorial taught how to post new data, delete data, and update data. These were pretty much the same as getting the data, so there wasn’t anything particularly new there.

After the methods were complete, the last part of the tutorial focused on chaining Observables and creating a search function. I learned about the async pipe, which is used for example when listing fetched items with ngFor. I also learned about the debounceTime(), distinctUntilChanged() and switchMap() functions.

After the tutorial was concluded, I can say that I prefer React over Angular. However, I liked Angular’s way of doing some things, for example, the lifecycle hooks. Unfortunately, Angular is very different from vanilla JS due to TypeScript and rxjs. This makes it difficult to get into and the learning curve is much steeper.