

LUT University
Software Development Skills: Full-Stack, Anytime Course
Name: Ville Saloranta

Exercises:

[https://github.com/CaptainCluster/
SoftwareDevSkills_FullStack_Modules](https://github.com/CaptainCluster/SoftwareDevSkills_FullStack_Modules)

The main project:

https://github.com/CaptainCluster/SDS_FullStack_Project

27.5. Enrollment on the course

Today marks the day I join this course. I intended to complete the devops course before this one, but an issue with Microsoft Azure, outside of my control, interrupted all the progress. I have already utilized multiple front-end and back-end tools in the past: *ReactJs*, *ExpressJs* and *Sass* especially. What I have not done yet is combining both of these and creating full-stack projects. This is something I have wanted to do, thus I am highly interested in what this course provides.

Out of all the libraries in a MEAN-stack project, *Angular* is the only one I have never used. I am curious and wish to get to explore its capabilities. In addition, I look forward to improving my skills with *MongoDB* and *Express*.

6.6.2024 Module 1 – NodeJS

I have already had NodeJS installed a long time ago and would prefer to use the newest version of it, as it would be inconvenient to go back and forth with multiple varying versions.

While I have used NodeJS, I have never used it via a terminal like demonstrated in the video which I thought was a cool thing and shows how flexible it is. Another thing that I learned was that devdependencies is a standard way to include dependencies that are only used for development. Prior to this revelation, I have just

installed everything as basic dependencies.

Seeing the demonstration regarding file manipulation and creation was a great way to further demonstrate NodeJS for me. Even cooler was the logger system, as it looked like something I could perhaps utilize in my own personal projects, and possibly in a work environment.

I was impressed by nodemon, as I have never used it before and saw how it made restarting a running server something that can be avoided. Creating a dev command for this could be a new habit for me when creating node.js projects in order to improve workflow and maximize convenience.

In conclusion, I especially enjoyed building the http server. It felt significant to me and I am happy with the learning outcome, as it strengthened my passion towards Full-Stack development. I look forward to doing more Express.js by completing the next module.

7.6.2024 Module 2 – MongoDB

Seems that I got too excited with ExpressJS that I forgot MongoDB is the next module I am going to undertake. I used MongoDB in a FreeCodeCamp course called *Back-End Development and APIs*. What I remember is that using it was quite straight-forward and recommended the use of a .env file for the inclusion of a cluster.

This module was quite short, but did show off plenty of essential commands. I chose to write them down while watching the video. In conclusion, there is not much that I have to say other than that I am confident in my ability to use MongoDB due to both past experience and this video.

7.6.2024 Module 3 – ExpressJS

After completing the MongoDB module, I finally got to the part where I get to use ExpressJS. Learning more about the freedom of ExpressJS felt very valuable, as that is one of the core defining aspects of this library. Despite the freedom to structure projects

in various ways, going through this module helped me understand what a more standardized way of handling it would be.

After building the member management site, I feel like I have learned a lot. ExpressJs is definitely a tool I consider essential for myself. On top of that, Handlebars was a fun tool to get introduced to. I also ended up downloading Postman, as I felt that not doing so would just be me causing myself harm, as it makes inspecting requests a lot more convenient.

8.6.2024 Module 4 – Angular

I have never used Angular before, thus this module will introduce me to something completely new to me. I have seen React, Vue and Angular as a trio for front-end development. Out of the three, I am only familiar with React, thus Angular could offer a decent opportunity for me to broaden my horizon.

One of the first things I noticed was that I am using a newer version of Angular, and thus I have to keep my eyes open for any issues that might arise. I tried to go back to the Angular 11 which was used in the tutorial, but I ran into issues with a bunch of npm packages, thus I saw the newest version as the best one to go with, despite the significant changes.

While I hope individual aspects are solved, I moreover hope that all the successful parts blend together and create the same end result as in the video. Figuring out how the things are done in newer versions of Angular, while slightly inconvenient and quite time-consuming, does seem to provide a reasonable opportunity to learn Angular, as just following a video is not enough to complete this module.

10.6.2024 Module 4 – Angular Part II

Today marks the day I have completed the fourth module (Angular). I managed to get around all of the issues regarding version incompatibilities. I figured out how to handle the imports and routes by myself, for the most part. The video did have a helpful comment section that had a few comments helping people with some

complex parts where the instructions in the video do not align with how newer versions of Angular function. After completing this, I am happy to say that I am ready for the final project. I anticipate that it combines all the tools of *MEAN-stack* which I look forward to.

10.6.2024 – MEAN-Stack Project

After completing the Angular module, I began working on the MEAN stack project. I went through the first three videos and did everything according to the video. The only difference in my approach was that I started a MongoDB cluster on the mongodb website and put the link to it in a .env file. This is, in my opinion, a comfortable way to establish a connection.

I used postman to test the results by sending a POST request. I initially got errors because saving no longer accepts callback as a parameter. After I fixed this error, I went to look at my older GitHub projects to see how to solve a problem where my app does not behave the same way it does in the video. I realized that while I did not get the same response, the data was still stored in the database.

11.6.2024 – MEAN-Stack Project Part 2

Today I began coding along by starting the fourth video. The videos are seven years old at the moment, and I can see that the presenter is, while talented, less experienced than in the newer videos. This has given me a good opportunity to watch the video and make my own improvements into the code.

I've been transitioning from K&R bracing where the left brace is on the same line as the function declaration to Allman bracing style where both braces have their own dedicated lines. Implementing this into arrow functions has been a gray area for me, but I started using Allman for them as well. Not only that, I have begun writing stronger documentation for functions, as I believe it will be a necessary trait in the future.

When I got to the Angular part, I chose to install an older version of it that aligns with the video series. I had to download

both older NodeJS and npm to get Angular 2 installed. Even after this, I had issues compiling Angular, as TypeScript packages had issues. I later on encountered issues with the Mongoose version I had. I managed to fix this issue as well, and from this point onward things have gone smooth-ish.

I am happy the original project has its source code available on GitHub, thus I was able to check the versions originally used in its package.json. This reduced many of the conflicts, while some still remained. I had to use my own knowledge to solve the remaining issues. Fixing all of this was time-consuming, but I think that being able to manage versions like this is essential, as I am aware some enterprises might use legacy libraries and code.

12.6.2024 - MEAN-Stack Project Part 3

I spent the entire day finishing the project. Today I did the final parts and managed to build my Angular front-end. This created a *public* folder that makes the entire *MEAN stack* application works via port 3000.

I spent most of my time trying to fix an issue with the JWT token authentication, as I was constantly given error 401. To solve this, I checked the package.json in Brad Traversy's GitHub repository (<https://github.com/bradtraversy/meanauthapp>) and copy-pasted some of the versions to update my Angular front-end side. With this done, I was able to use the HttpClientModule that is newer than the library in the older version I used. I managed to write the code in a way that functions with this library and chose to go with it, rather than libraries such as Headers.

Although I used HttpClientModule, it ended up not being the ultimate solution to the error 401 issue. What caused the issue was that the router POST request to /authenticate in users.js was not precisely the same. I spent many hours without noticing this, but I was happy to learn more about JWT, HTTP headers, etc while trying to fix the issue. In the end, trying to solve this made my code stronger and enhanced my knowledge.

12.6.2024 – Conclusion

Now I have a functioning MEAN Stack application. I am happy with my learning outcome. I originally felt doubt about learning by coding along while watching videos, but I feel like I learned a lot by watching all these videos. Angular is a great tool. I already have a full-stack project in mind that I want to start working on during my free time, thus now I am debating whether either Angular or React would be more fitting for it. These kinds of projects are why I was initially interested in this course.

I feel like I learned everything For example, I feel like I need to know now. File structuring was a big question for me, but now I believe that I know what to do regarding that. While handling various versions was moderately inconvenient (partially in a good way as it forced me to explore and learn), I still enjoyed the course and feel like all the time I spent messing around was worth it.