Lappeenrannan–Lahden teknillinen yliopisto

School of Engineering Science

Software Development Skills

Jani Heinikoski, 0541122

LEARNING DIARY, FULL STACK MODULE

**LEARNING DIARY**

22.07.2022

I started by carefully studying the course Moodle page. I learnt that the main technology stack used in this module is the so-called MEAN-stack (MongoDB, Express JS, Angular JS and Node JS). I have previously completed the mobile and front end modules of these SDS-courses so the Moodle page was very familiar. I noticed that this module follows the same paradigm as the other two modules I completed (studying and doing exercises/coursework, writing a learning diary and finally demonstrating my new skills with a unique project).

I began by creating a new Git code repository for this module. I took the opportunity to learn something new about git as well by watching YouTube videos and reading Git documentation about branches. I might utilize branching in Git to test some features of the project in their isolated branches before merging them to the main branch. However, I already had Git installed (specifically GitHub Desktop -application on Windows) so there was not too much to learn when setting up the code repository. Furthermore, I have already setup Visual Studio Code (my go-to code editor).

I continued to the module’s task list and started at the first one called “NodeJS”. I noticed that there are two videos available: an updated video and an old one. I took the liberty of assuming that only the updated video needs to be watched. The first ten minutes of the video focused on talking about Node and npm on a general level both of which I am already familiar with. Although, I did learn that Node has been written using C++. The non-blocking nature of Node, its single threaded event loop (or rather the JavaScript event loop) and also the Node package ecosystem were familiar concepts to me so the first ten minutes was mostly a recap of the main points about Node and npm.

Next part of the video concentrated on installing Node which I had already installed. It was recommended to install Node version 12 in the course Moodle page so I just used nvm (Node version manager) to switch my current version to 12.22.6 (which I had already downloaded using nvm).

Next up I initialized the first module’s working directory with npm init to be able to install third party node packages. I installed nodemon as a “dev dependency” even though I have it globally installed on my machine. Then I added a “gitignore”-file on the coursework folder to ignore node\_modules folders and package-lock.json files in all subdirectories. Again, installing dependencies (other node packages) was familiar to me.

Next, I coded along the simple examples shown in the video. I did not learn much but I did not know about the module wrapper function which wraps the node modules before runtime giving them access to variables: exports, require, module, \_\_filename, and \_\_dirname (which I previously have used but had not thought why they are accessible).

Traversy Media (author of the Introduction video) briefly brushed on the topic that Node did not support ES6 modules (specifically the import/export syntax) at the time of making the video. I always found this topic very confusing because I have stumbled many times on problems with mixing CommonJS module system with the ES6 module system. I Googled for a good minute about the topic and from what I gathered Node has supported ES6 modules since version 12 and has become the official standard format to package JavaScript code for reuse. To tell Node to use the ES module system, the type property can be used in the package.json file or the .mjs file extension can be used.

Next topic in the video was about the Node’s core modules.