Lappeenrannan teknillinen yliopisto
School of Business and Management

Software Development Skills

**Johannes Hölker, Studentnumber: 609983**

# LEARNING DIARY
# SOFTWARE DEVELOPMENT SKILLS
# FULL STACK
# MODULE

# Contents

# 1 20.10.2020

Dear Diary,
After reading the introduction of the course, I replicated the Word Template for the diary in LaTeXusing Texmaker as a GUI for writing. I managed finding my old Github Account then and installed Git on my Ubuntu system. I decided using Fork for my version control because I used it before and I am familiar with the program.

# 2 22.10.2020

I wanted to download Fork but I realised that Fork (and SourceTree) don't have a Linux Version. During watching the Introduction to Version Control I saw that the code editor Atom has an implemented version control GUI, so I decided to take that one.
Afterwards I watched the video about Node.js. Because I planned improving my Python skills this semester I realised that the course is made for JavaScript, which I didn't use before so it is a whole new challenge for me. Regarding the point that JavaScript is useful for sending/getting data from a server, I am looking forward to it.
I already have an idea for a JS application. With a friend of mine I was planning to make a simple ToDo-list application using the Python. For storing (and checking) the entries of the list on a server, I am thinking about using JavaScript.
Now I am setting up my Git-Repository and placing the Texmaker-Workspace in there. I tested the first commit of the folder $SD_FS$ in the command line and the second commit directly using the extension of Atom. During this I am getting familiar with the code editor Atom and the Github implementation in it.

# 3 26.10.2020

During watching the video about Node.js I am creating my Exercises Workspace and the Project Workspace in my project folder. After I tried isntalling Node.js by using the tar.xz folder, which i didn't managed, I nstalled it using the command:
sudo apt install nodejs
After doing so I realised that I installe the version 10.19. So I deinstalled the program and installed it using PPA. It didn't work either. So I used the version manager by tyoing the following commands:
$curl-sLhttps://raw.githubusercontent.com/creationix/nvm/v0.31.0\,install.sh-oinstall_nvm.sh$
$bashinstall_nvm.sh$
nvm install 12.19.0
And setting the version to default.

## 4 27.10.2020

My environment during the Node.js Crashcourse contains now Atom and my Linux Terminal and it works all fine while the exercise till I wanted to use quotes in a String. The quotes with the dollarsign were not recognised as a quote. I started looking it uo and also started the JavaScript course on Codecadamy for learning the Basics of JS.
I figured out that I used the wrong quotation marks. Using the ones from the video my quotes are working fine.
In the end of the Youtube video I made myself a Heroku account and got my first Website online. It was really exciting and I am looking forward to my project now. You can see the result following this link:
https://blooming-fjord-46718.herokuapp.com/
After finishing the video I made a Commit on Github. Here I had the problem that the subfolder of SD FS was not committed. I looked it up.

## 5 31.10.2020

In the git documentary I saw that you need submodules for committing to two public repositories. So I decided to avoid these circumstances by making two project folders. One for Heroku and one for Github. I made myself a more serious Github account and seperated the folder for Heroku from it. Before committing I am making copies of the Heroku project into the Github project for sharing the code. I will also exclude the LaTeXcontent and will just make commits with the Diary pdf file in it.
After I setted up my new project folder I started the video about MongoDB. I installed the Linux version and eberything worked fine till I wanted o use MongoDB compass. It said to me the connection to the default server was refused. I shutted down the computer and quitted for today.

## 6 01.11.20

After the startup of my computer I tried the same thing as the previous day and eventually MongoDB was running. I didn't make any change. So I can continue with the MongoDB crash course with setting up the databases.

## 7 02.11.20

During the MongoDB Crash Course I experienced some very useful tools which are perfect for a ToDo list application that I thought I can build. In the end of the video I saw that Brad Traversy did a To Do entry and I definitly thought about doing this in my application.
I took a small look into the Example project Video series and I realised what the application should be about. I planned to do the example project too for

getting a Login page and then to implement a To Do list for every user.
It is especially really useful to store the ToDo entries in a MongoDB server for accessing and changing them from several devices.
Short after I started the Express Crash Course and setting a local server I made a commit on Github with all the tutorial stuff.
I continued with the video and everything becomes clearer for me. Especially the part of Express that you can just make .html files in one folder and a new page is opening amazes me.

## 8 03.11.20

I continued with the Express Crash Course and everything worked fine. After finishing the video I made a commit of my work and pushed it to Github.
I am now thinking about my own project and I searched for some inspiration in an Ted interview of Linus Torvalds. He says that his purpose is to make applications he needs. Here is therefore a collection of ideas of a garden/plant organizer. The functions that the user should be able to do on the website are:

- creating and logging into an account
- accessing a list of vegetables which is public and which he can extend
- adding vegetables from the public list to hiw own plant list
- creating hiw own list of vegetables he wants to plant next
- receiving a list of activities (like a ToDo-list) he has to do next
- also the weather should be displayed on the front page/Dashboard

By using the given Tutorials the application should be able to do the following:

- storing the entries for vegetables on a seperate server (MongoDB)
- the user profile should be protected with a password
- the Dashboard should change depending on the weather (also the weather of the last days) and the special season (written in JavaScript)
- having a handy and easy to use surface (Angular)

The vegetable Object should have the following:

- Caption
- description
- harvest time
- plant time
- pouring frequency
- optional Notes

## 9 04.11.20

I started the Angular Tutorial. While setting up my environment I decided to directly code in the git controlled dircetory this time. I setted up my environment again while using Atom (which I got failiar with and with which I am making my Commits and Pushes) and my bash Terminal. I got the default application from angular prompted in my browser.
I faced a problem while in the first section. I was not able to refresh the browser that it showed the name of the hero. I restarted the whole Tutorial project and tried it several times while I tried to understand what the Angular framework created there. I understood that it used Node.js and it wants to simplify the creation of a good looking website but I needed some time to get what the Initialisation created.
One thing that I experienced was that the .css files changed to .scss files. During the whole project up for one more time I realised that I chose the SCSS option in the initialisation.
In the third try the application of the first chapter of the Angular Tutorial everything worked fine. I had to restart the "serve" command because before restarting the input was not displayed above. I couldn't figure out why. After finishing the first chapter I made a Commit of my work.

## 10 05.11.20

I continued with working on the Angular Tutorial now doing the fourth chapter about how to add a service. While doing this I checked (way too late) what the language TypeScript is. I experienced like I guessed before that it is nearly the same a Java Script and that every JavaScript file can be compiled as a .ts file. During the process I made a safety Commit. I also trained myself further in JavaScript by reading over the basic commands.
After finishing the fourth chapter I understood the basic concepts of prompting messages, but it is still a long way to understand every command. In the fifth chapter (in-app-navigation) I hope to see better where the commands are connected to the outcome in the application. I am now really looking forward to start my own project to face this problems and to learn it better. I made a commit with my previous work.

## 11 07.11.20

I changed my way of learning the Angular Tutorial and Javascript/HTML in general. I am now going through the last chapters of the tutorial writing comments to all elements of the project. With changing names and testing different code I am tracking the outcome with the "ng serve" command process running in terminal. The commands with a questionmark mean that I am not totally

sure. After I understood the basic structure of the HTML pages and found every call for them a commit was made.

## 12 08.11.20

During the fifth chapter of the angular tutorial, I had a problem with routes. My Angular CLI displayed a problem in the file app.component.html. I couldn't figure out what was going wrong so I checked out the latest commit (commented commit). First I tried to store the .html files in their old directories (because I changed it the last time for my own benefit). After the problem came up again I saw that the routing files existed before the commit. I deleted them because I thought it maybe causes problems that they are not generated right by the CLI and I was right. After generating them again everything worked fine.
Before going on with seperating the details view in its own page I made a Commit (third safety Commit)

## 13 10.11.20

For reviewing the whole Angular Tutorial I am setting up my own Project development environment following the Tutorial once again. At the same time I add comments to the Exercise for better understanding what I am doing. THis is a first test and I am not sure if it works the first time.

## 14 19.11.20

The angular tutorial is finished now. I made the sixth part (Adding/deleting/Search function) in the train to Rovaniemi. Everything worked quite well and I made a commit with the train wifi at the moment when we saw the first Finnish snow. I started watching the Example Project playlist afterwards and I am looking forward to code along.

## 15 21.11.20

For the beginning coding along the video series and creating my own project seems to be the best choice. So the next commit will content a new application folder in Project Workspace called gardo. In there the modified Example-Project will be developed.

## 16 26.11.20

In the third video of the playlist a problem with mongodb occured. In my Mongo Shell (which I am using in a Linux Bash Terminal with command "mongo") I cannot see the necessary db server called "gardo". I didn' do the request on

Postman and realised that the db server is ony created after sending a request. However the problem persisted (I got the error "TypeError: User is not a constructor") in Postman and the nodemon shell at the same time) and I checked my code from the third video. After a while I found a feew mistakes and with the right code and a local request using postman, I got the right respond.
After beginning the fourth video I saw that my nodemon engine is not working. After typing "npm start" a deprecation warning occurred. Also using command "nodemon" the same thing happened. I watched on StackOverFlow for the problem and I found out that there are some version problems. Adding the following in app.js helped fixing the problem.
mongoose.connect(config.database, useNewUrlParser: true , useUnifiedTopology: true);
However this didn't work but the command "nodemon" worked fine. Instead I have another error probably in passport.js. It says: "Error: Expected "payload" to be a plain object." when I am doing an authenticate request with postman. I tried several things but nothing worked yet. I checked my whole code and it is completely the same than in the video.
Finally I looked in the source code on Github and saw that Brad had data:user instead of user in the ".sign" function in users.js. After changing it I received the token.

## 17 29.11.20

Going on with video 4 of the Full-Stack playlist of Brad Traversy I used the received token for a get request to test the authorization of a specific user. Eventually I had the same problem than in the video but the solution didn't work. I still got "Unauthorized" so I tested the code using "console.log(jwt_payload._id);" which idn't gave me the expected id. Instead "undefined" appeared into the console. Trying "console.log(jwtpayload.data.id);" gave me the expected ID number so I used this one and I got the right resonse in Postman.
In the beginning of Video 5 I uninstalled the actual version of the angular-cli and installed the old version as told in the course description. When I want to run "ng serve" in my new angular-src folder an error appears:
"ERROR in /Project_Workspace/gardo/gardoapp/angular-src/node_ modules/@types/node/index.d.ts (20,1): Invalid 'reference' directive syntax." After I searched for this error with google I found an explanation on Stackoverflow (https://stackoverflow.com/questions/63720450/node-modules-types-node-index-d-ts20-1-error-ts1084-invalid-reference-dir). I changed a command in the index.d.ts file (// ¡reference lib="es2015" /¿) in node_modules and my angular project compiled successfully.
I made a commit of this and went on with the video number 5. After writing the exact same navbar.component.html than in the video I got a crashed website with just MeanAuth App to read. My suggestion is that the bootswatch version changed and that the html commands are not working anymore.
Regarding this I will use the HTML code from bootswatch.com and will integrate the content of the video.

## 18 30.11.20

After completing the video 5 (I couldn't manage placing Login and Register on the right side of the navbar yet) I made a commit and went on with video number 6.
In Video 6 I improved my skills with implementing the flash messages and everything went fine. I made a commit after that video and went on with video number 7.
After creating the registration form and a successful registration of a test user I made another commit and went on with the next video.
Storing the specific user token in local storage and watching it in the web console was very interesting.
In Video 8 everything worked fine as well as the half of video 9 where I made another commit.

## 19 01.12.20

In video 9 I made a few mistakes but I found them and finally my Authentification of the Gardo App works and the dashboard and profile are only accessed by a logged in user. I am postponing the deployment to Heroku to the end of my project so I can develop my own features now.
In the beginning I want to add a public list of vegetables which every user can access and can add entries to. This list should be visible on the dashboard as well as on its own route.
For the beginning I am going once again through the Angular Tuorial for creating the public Vegetables list.
Once the vegetables list is created I made another commit. While going on with the Angular Tutorial I am exeriencing an error: "node_modules/rxjs/Rx"' has no exported member 'of'." I was confused because everything was the same than in the Angular Tutorial. After googling the error message I got the anser that the import works only for Angular 6. While I am using Angular 5.x I had to use another import in vegetable.service.ts. Once I did this I got no error and it worked fine.
No nothing worked. I got a strange message in the backend server using nodemon : "5fc4b157a788e735f02ffb1b". It happens every time when I am clicking on List of Vegetables and this error was not there before I implemented the Observables.

## 20 06.12.2020

As mentioned before I had a problem with the Angular version (because in the example project Version 5 is used). Therefore I deleted all changes since the last commit and tried implementing the Angular Service "VegetableService" once again but this time following the instructions of Angular v5. Eventually I got my Website working and now I want to go on with the Angular Tutorial to

build a public list with Http request where every user can add entries.
Because HttpClient from @angular/common/http (like it is explained in the Angular v5 tutorial) is not found after implementing, I am thinking where I want to save my public list. In my opinion the best way is to store it alse on the MongoDB server currently running on localhost:3000. I will try doing the actual angular tutorial again with implementing the HttpClient and storing the list directly on the MongoDB server.
I realised that the Angular version of the Example Project is even older. So I am following the Angular Tutorial of V4 now and I hope that there are no complications.
Alright I am not following it in order to skip implementing fake server. So I started adding a second collection to the backend server. I just had to do the same as with the users collection. After some testing the request through postman worked.
Following the structure of Brad Traversy I implemented three functions on the Backend server.

- addveg: for adding a specific vegetable to the vegetable collection
- seeveg: get one vegetable and its data by name
- seevegs: get all vegetables instantly

## 21 07.12.2020

Trying several different ways (e.g. using toPromise()) didn't work yet. The main problem is that I don't know exactly where to start. Shall I start with the Html form of adding, shal I start with the function in the component or in the service or shall I configure the backend server first.
So I made a plan in which order I will go on. First I want to configure the backend server, then I am creating the Html and CSS files for my "List of vegetables" route including a new component "vegetable-adding" and then I will implement the fucntion in the "VegetableService" and the "Vegetable Component". I will do exactly the same as Brad Traversy did in the video when he created the register route.

## 22 08.12.2020

Following my plan from the previous day I finally created the adding form for one vegetable using the submit function. After some simple errors the app is working and I made another commit (Some other stages were committed during the process of coding).
In the next step I want to add a button under the vegetable-detail component on the list of vegetables route. The button shall redirect to a route /vegetables/planting with which the user can declare his Plant date and specific stuff for planting. On this route shall also be the vegetable-detail component and a

form. In order to make the vegetable-detail component running there has to be a variable containing the selected vegetable passed to the component vegetable-planting. I am struggling with this because I am not sure if I have to use a function or ust an exported const. None of it works so I made a commit with the stuff hich worked. This was the redirection and the added button. Also new components (garden, garden-planted) are created.

## 23 09.12.2020

Because I am running out of time and the redirecting didn't worked yet (because I didn't know how to pass a variable properly) I extended the vegetable-detail component with the whole planting form. Every time I am showing the vegetable-detail component, the user can add this specific vegetable to his garden. This was not plant like this but it makes things way easier. I made commit of the successful work. Now I want to store the added plant properly to the users information on the backend server.
First using the mongo-shell I figured out how to append something to existing data. I found the command "db.collection.updateOne()" and I read about aggregations in mongo. With the entry "$addFields" a field can be added. But it took me a while to bring query and aggregation in the right order and to understand how it works. Finally I found the right command which I now want to use on my backend server as a new function.

## 24 10.12.2020

Copying the "register" function and creating an "update" function for appending the user data was easy but it didn't work out. I tried several things in user.js and users.js which I created back for the authorization ability using Postman and nodemon. I tried for example beginning with the findOne() function (which didn't work because it was a Post request) and I altered the passed in data slightly but nothing worked. In the end I figured out that my input for the funciton should not be a new user (like in register). With passing in just the username and a new field finally it worked. So the new funciton is a combination of register and authenticate. After the succes that I now can append my userdata, a commit of the work was made.

## 25 11.12.2020

Before creating the "Plant this one" button and function, what I was planning to do next, the Backend server request for the users planted vegetables was made. I used the "getprofile()" function for this and extracted the vegetable json object out of the already existing collection users. After using some useful things out of the profile and vegetables components, I got the request working

also on the frontend.

## 26 12.12.2020

Finally after several tries I managed the backend post from the frontend. Every time a user clicks on "Lets plant this one", the specific vegetable is added to the "plants" array of the user data. For this I am using the updateOne() function as explained before. The updateOne() function requests the username and the plants array. Using the username the function is searching for the user and it replaces the plants array with a new one. The new one is created in the 20. Commit and it contains all old plants and one new plant.
So for now the mongo server is running properly and the basic functions are fulfilled. Obviously some important things are missing (e.g. checking for duplicate usernames while registering, deleting of vegetables from own list.) but in the next step I will go over the code in order to clean it up and to implement all necessary field.