Lappeenrannan teknillinen yliopisto

School of Business and Management

Sofware Development Skills

Maslov Ilia, x083784

LEARNING DIARY, BACKEND MODULE

**LEARNING DIARY**

20.06.20  
  
I checked the general information and read about all the four modules that I have enrolled into. In the past, I have been participating in the high school extracurricular activities to study front-end development, with a full cycle of such a development, with a few topics concerning the graphical design and the back-end programming. Ever since, I have been rarely employed directly as a software developer (even though, I did some websites for my friends for some minor remuneration). To be frank, I haven’t really wanted to become a front-end developer, but rather a programmer, but I was told that “I should take a path to front-end dev, and then may be, if you want, you would get into programming branch of the extracurricular course.” As such, I have spent 3 years of such 3 hours/week in a classroom (a predecessor of bootcamps, but for high schoolers). In the process somehow, I wasn’t entirely sure that I “really needed this,” but since I started, I have a habit of finishing something, sort of on the auto-pilot. Quite recently, I have found that the industry has become significantly larger, e-commerce and digital marketing expanded, and if before webdevelopment was for hobbysts, now it is quite fashionable even. I have combined my previous studies of webdevelopment with occasional MOOC courses. I have got a business degree, and IT management degree on my Master’s. Now, since I am slowly finding myself in the IT industry where I have a sort of talent, I (even if I don’t get to be a programmer), still might benefit as a manager with the knowledge from these courses. Hence, here I am, studying it.

In short, on the first lecture I have got to know the Git control (I heard of it before, but never used), and got to install it and run. I learned to set up a git repository and did my first commit, everything went smoothly after I clicked the banner to watch intro to GIT. I’ve also tried to set up my environment, and I decided to use Atom, as I have used it previously in MOOC (a power of habit that if you used once, you tend to use it always, eh?) I have also tried to do the first projects (but mainly, I focused on getting the grasp of the overall course content).

By the end of the day (after some good sunbathing from the home, as I happen to have a wide frame window facing the West, and chicha smoking session), I have got to listen to the first project about the node.js. The really cool things started when I got to write the first code in node.js to host a “placeholder server”. In-before, I have run such test servers, but via the app (don’t remember the name), which was used to do exactly the same. I suppose, there’s some fractional utility in knowing how to set up such servers. However, there was some pleasure in seeing actually (1) employing Git, which I didn’t study before, and understanding it better by practice, and (2) using a Node.js, which is somehow often spoken about by the IT professionals, but was somehow out of touch for me.

Now, I could see that the knowledge is closer for my understanding. Probably the most important lesson is that I have somehow touched on the “limiting beliefs” that “one has to be extraordinaly smart for it”, “must know a lot of math before getting to do it.” Even though I don’t see myself as a full-time software developer, I might enjoy an intermediate communicator position between the business and software developers, and for that I would like to know the topic of programming. Hence, limiting beliefs might be good to get rid of in the process of studying at these courses.

21.06.20

In the second day of studying backend, I have continued with a MongoDB. The syntax learning was somewhat hard, and I think it would be the most crucial aspect for me to learn, as otherwise I feel like I have a good grasp of understanding of the architectural principles of managing a MongoDB (ofc, at the level for a beginner-to-intermediate).

Basically, listening to the lecture was mostly about repeating the syntax operands, and also knowing how to set up the shell function, GUI browser and (as I did all of that in the beginning) managing the DBs on the free cloud. I would say that I would seek to use the MongoDB (if I find out how to integrate the Node.JS manipulations with the cloud server) to make the project. I don’t really know what that would be. However, at least the knowledge of knowing how to set up some sort of a server and manipulate with it over a local machine seems like a cool skill. I always thought of something similar like very hard, but it seems not that much out of reach, at least for the purposes of a single individual I think these would be enough to know to m.b. make a simple showcase app and m.b. get an internship in the future to develop the skills further. Furthermore, I have again practiced some git pushing commands, but based on memory.

22.06.2020

The next day in the beginning I thought of starting to continue the lecture. I have done the one-in-one code as was shown in the video. And then at the moment when it was supposed to work, I got some bizzare error message that db is not defined, although I have checked and it must have been working fine. As I have got no clue what are all of these syntaxes mean, I got really annoyed that this is very bad teaching. Similarly to the aborigines trying to remake the wooden plane after the Americans left the Indonesia in the Vietnam war or something, I have got no clue what these mean. And the crash course person (obviously he knows the thing) doesn’t explain and just says “do this, do that”. I don’t expect to have everything on the plate, but this is very bad – it forces me to go through the basics on my own – hence rendering the whole crash course by itself useless. What’s the point of doing something if you don’t understand the thing? Ok, in the mongodb, I have got to understand it because I have gone myself in the past about the MySQL and understood it there. In here I too understand logically but very roughly what are the elements’ interactions with each other. However, it is not explained. I could even see that there’s an express-validator of a specific version that must be installed (which I did), and I also see that the problem is in this area. But this is not good. I mean, there is some utility in this, but it differs little from the “go figure yourself,” the point of the teacher of the course then dissipates and it’s a tango one-in-one. All of that mini-rant is well shown in the joke video below, it has an 80% resemblance actually to this dude who’s referred to by the link from the course. I understand now that the problem was about the express-validator, but overall I have had to find the solution in YT comments (and the preemptive note in the spoiler in the course content didn’t do good explanations about warning of the rakes in the study content). Otherwise, I think the syntax content is explained poorly in this lecture, and even before – all js is not explained as to what functions, vars, const and objects are (even though I understand overall what these are, but even I with some previous programming experience didn’t manage to get that info, I wonder what would be for those who didn’t even program…) To be frank, all of that was very discouraging introduction to the backend.

<https://www.youtube.com/watch?v=Rp9e1Y-vdBM>

Just for me to know how to setup the git:

**#CAPITAL LETTERS ARE IMPORTANT**

1. Create a repository on bitbucket
2. Git init
3. Passwords/Login
   1. git config --global user.name "Ilia Maslov"
   2. git config --global user.email 'eliasmaslow@gmail.com'
4. [**REPLACE**] (how it will be called) + (connect link)
   1. git remote add SDS\_ProjectFinal https://TrTrwerwerTrueElim@bitbucket.org/TrueElim/sds\_be\_finalproject.git
   2. git remote add origin https://TrTrwerwerTrueElim@bitbucket.org/TrueElim/sds\_be\_finalproject.git
5. git clone from bitbucket
6. git commit -m ‘First setup git’
7. git push --set-upstream SDS\_ProjectFinal master
8. + EXTRA COMMANDS:
   1. git branch qweqwe
   2. git pull origin master

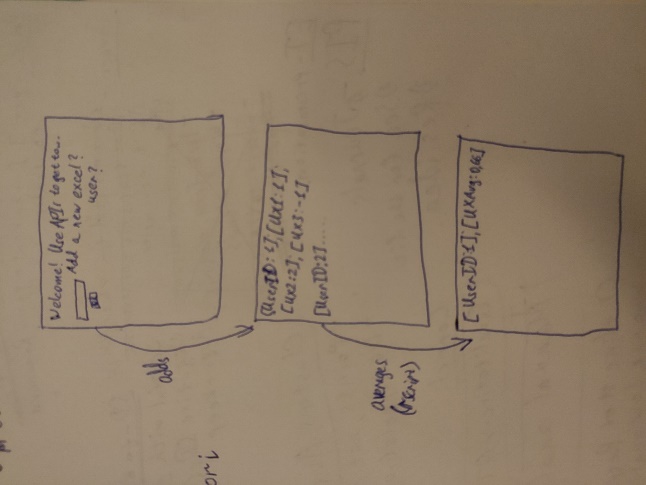
**THEN, to start the package.json and app.js:**

Npm install [package names]

Touch app.js

Then in the end of the day, I have decided to go myself into the idea, as (at least something about server setup and how to make it that the code could be combined with other elements) to make the project of data analysis with MongoDB.

The conception stage was in-before: I have decided to make a simple three-page project (as I count API pages as a page), whereby a user could upload data (in hand, or by excel?), and then by pressing to process the script could get another API page with the results of churning out from the script.



23.06.20

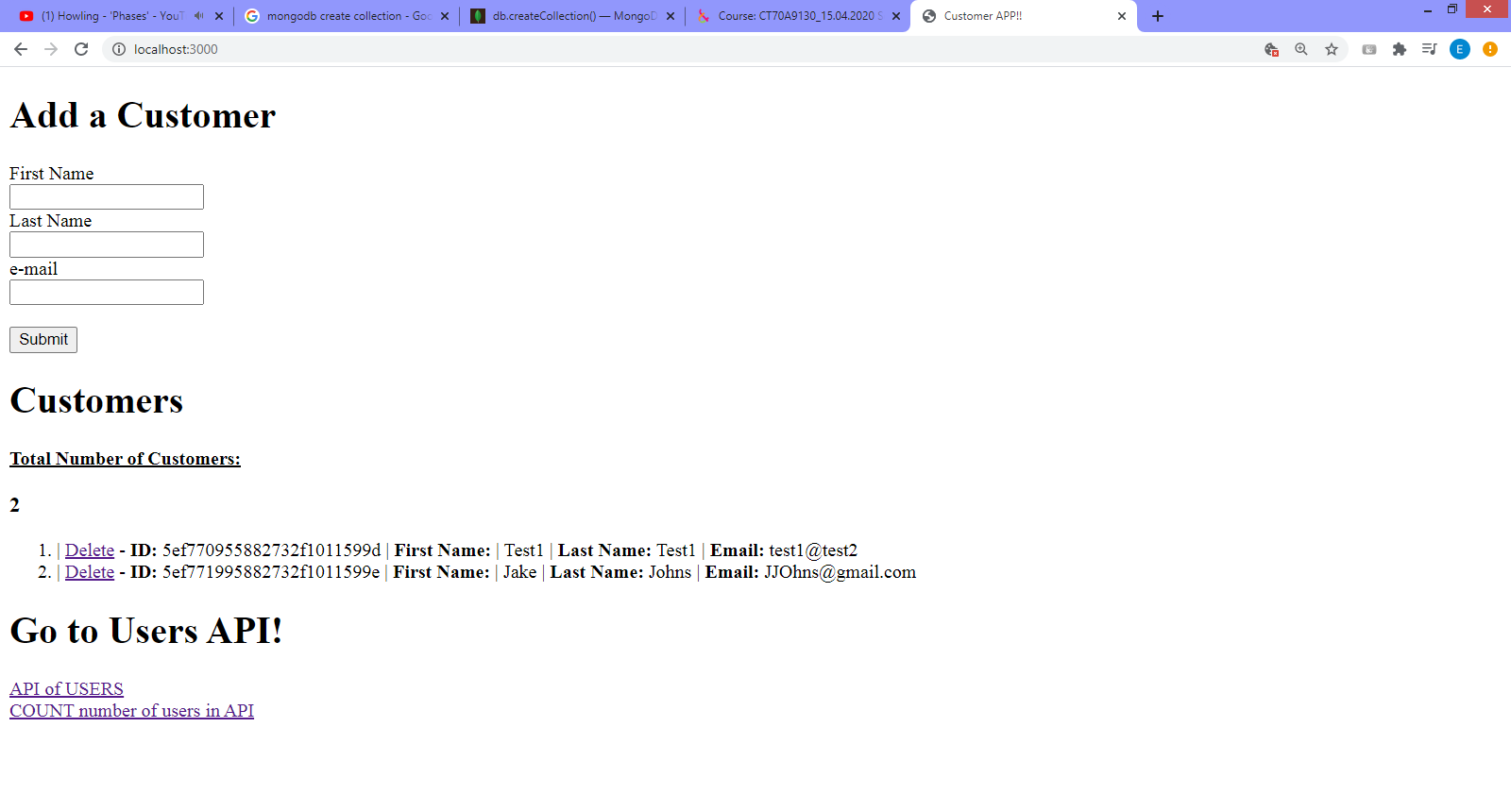
On this day I have moved onto the final RESTful API ME\*N framework (no Angular, only backend).

* **Resteasy – used as a tool to view the API/genres in a GUI – CRUD operands to interact with the data, as if resteasy is a side API that send commands etc. to the app that I am making**

To operate and to program, I have ran a git according to the 1-9 steps (shown above), hence I managed to make a remote git specifically for that project separately. Then, I have followed the instructions and these were easier to do so than the last time as I already knew more of the objects, functions, etc. Still, I didn’t understand the syntax, but I kind of understood what the code stood for, and managed to find the errors, as they arised for me. As a result, in principle, I have accomplished all of the required tutorials for the backend course, and actually look forward to make a project. As I spoke with the teacher over email, I was reassured that the project mustn’t be something very complex, and somewhat tweaked of the original code, as it’s 3 Cr., but in principle I would have liked to do my initial idea. However, I think it would be too much of learning syntax just now, given that I also work as a salesman in B2B of the software, I couldn’t devote a lot of time to learn it all that I would do such a project. However, in the future, I could potentially try to (if I get to see other similar courses, or would get the basis from which I would continue learning on how to do my initial project). I would say that this initial idea for the project is interesting, as for me I am also employed in the software of data analysis, and there’s a similar problem – how to get the data, analyze it in R, and then make an output to the user. Anyway, I would proceed to making a simpler project by studying to begin with some syntax and understanding the overall structure of JS.

27.06.20

On that day I have done my project and finished it. For this purpose, I have decided to use the example code from the last two sections, in a way combining the two. I have made a project in which it is both possible to add a new customer (first/last name/email), to see the customer names, all of which is facilitated with the use of mongodb. From the front page I made it possible to remove any customer. Furthermore, I have referred to the API, so that it is also possible to retrieve APIs of any specific customer. Furthermore, I have added a side feature so that I calculate the API of the count of all customers, and then extract that API into the main page (where I could add/remove customers).



Link to the video showing how the project works:

<https://www.youtube.com/watch?v=ZfMZnmIngPc>

It is really simplistic, not up to the scale of my initial idea, but I have decided to indeed keep it simple, as asked by the teacher. As a result of it, I have started to understand and get a good grasp on what’s an API and what does it mean to create a web application with API architecture imbedded into it. As for the previous issue with the code, I would have to admit that there are certain skills that I have acquired from the project, that’s for sure. Furthermore, I believe that I would be able to get that project as a showcase to support my business studies profile. However, I still think that javascript fundamentals is still missing from the course, which made it harder to understand it. For example, I only later when I did a project understood what <% %> meant (a sort of space for including the .js code). Furthermore, I didn’t exactly understand what different chunks of codes meant (until I practiced it a little in my project, which still wasn’t particularly clear). For example, things like why to perform CRUD operands from the express tutorial one has to create two separate functions: one function in the genre/book, and another in the entry point. These things were left out of explanation, and I have developed my intuitive understanding before the course, which helped me to somehow get a grasp on it. In general, the course was fine, and I am glad that I sticked to it in order to finish it and made a project. In that regard it was a good practice and good satisfaction that I have set out a task to achieve and I did in the end achieve it.