Lappeenrannan teknillinen yliopisto

Software Development Skills

Binghui Liu, *0597037*

LEARNING DIARY

**LEARNING DIARY**

**Node.js**

12.5.2020  
  
I learned the basic knowledge of Node.js. There are some features such as single-thread, non-blocking I/O calls that differ it from other techs like PHP. It’s built on Event-Loop to handle concurrency. I also learned about some usage scenario for Node.js. I installed Node.js v12.16.3.  
I learned what is npm and how to use package.json for package management. I also created a git repo on GitHub and set up VSCode for Node. After that by following the tutorial I created a simple web server and commit the changes.

**MongoDB**

13.5.2020

I installed MongoDB via homebrew on Mac OS X, the process of which is a bit different from the instruction meant for Windows. I also learned the basic syntax and operations of MongoDB.

**Express**

14.5.2020

I installed Express.js and Body-Parser for middleware and path parsing. I also installed nodemon and attached VSCode debugger to it. Then by following the tutorial, I installed ejs for using view templates, and created a form input.

15.5.2020

I added mongoDB support for data finding, insertion and deletion. Though without any proper css the webpage looks quite ugly. I also followed the tutorial to add jQuery for interaction. It wasn’t very clear in jQuery context what is “this.data().id” or how to get the data-id of an object but I got the answer from StackOverflow.

**Angular**

18.5.2020

Today I restructured the git repo directories. But after moving the Node.js I found I need to create a new launch file for VSCode. It’s also important to remember to start the mongodb service before running the app.

By following the tutorial, I initialized an angular workspace, updated the boilerplate to display a list of heroes from mock data, and also added selection event.

19.5.2020

I added a new component to the workspace which makes the structure clearer. I also created services for dependency injection. The concept of dependency injection was quite hard to understand in Android development, but it’s easier to catch in Angular. It’s useful to know event binding also applies for service.

20.5.2020

The routing module was created when the workspace was initialized, so I just need to add the routing paths. It’s also not easy to understand the parameterized route concept, especially how the get the parameter from the routing snapshot.

RxJS and Observable are two important concepts in Angular. Observable is relatively easier to understand as it stands for asynchronization. RxJS has a weird syntax compared to normal JavaScript.

**MEAN Stack**

21.5.2020

By following part 2 of the tutorial, I set up express and routing. First time it failed because when importing users module, the path was wrong. It was “../routes/users” instead of “./routes/users” since routes folder and app.js are at the same directory level.

By following part 3 of the tutorial I created User model and successfully created a new user entry in mongo db via Postman.

22.5.2020

Today using I learned how to generate json web token using passport.js. When following the tutorial, I got an error “Expected "payload" to be a plain object. at validate” when using jwt.sign, and I found the solution here

<https://stackoverflow.com/questions/47117709/payload-error-in-jsonwebtoken>.

It is because the user returned by mongoose query is a mongoosejs object, so it needs to be converted to Json object. Though the route protection took me quite a while. Somehow the GET /profile never hit the passport jwt strategy. After an hour I found the reason was because when generating a jwt token, in my code I forgot to put a whitespace between “JWT” and the strings behind it. After checking the sample code I found this bug and fixed it. Then the code worked as in the tutorial.

23.5.2020

I initialized Angular in the app. It was helpful to do the Angular tutorial beforehand, so now I have a quite good understanding of the concepts such as component and routing.

When using bootstrap for navbar, the process is different. Now with Bootstrap 4 navbar items alignment methods are different, and one biggest change is float-right won’t work since it’s flexbox. I found the correct way to do right alignment from the link below

<https://stackoverflow.com/questions/41513463/bootstrap-4-align-navbar-items-to-the-right/41513784>

24.5.2020

Adding validate service is quite straightforward. But I spent some time trying to figure out why the submit is not triggered in the register form, since I didn’t see any console output. In the end it turned out to be I was looking at the wrong place: I was checking the console output of the node service, i.e. the backend. Instead, I should be looking at the browser console which is the frontend.

25.5.2020

Today I created the register function. Since the tutorial is a bit outdated, the http module in Angular has changed, as well as the response which by default is a json object. Instead of getting data.success, it has to be (data as any).success, because data is an Object, the type casting is required.

27.5.2020

I implemented the login and logout functionality. It seems that when a component is updated sometimes angular server needs to be restarted.

28.5.2020

Today the app dev tutorial is complete. I added route protector via angular jwt. The angular 2 jwt in the tutorial is an older version, so basically it means the module can handle jwt token loading from local storage by itself. So in the end an extra loadToken() is not required. The last step is to run the build script and build the website to the public folder and then it’s done.