

Research Document

Authentication and Authorization

Fontys University of Applied Science Software

Project Plan Individual project:



Henaknowledge

Tutor: Tim Kurvers & Maja pesic

Student: Mohammed Al Harbi

4089553

Contents

Introduction	3
Research question:	3
The required criteria for completion	3
Results of the research:	4
Step 1:	4
Step 2:	5
Step 3:	6
Conclusion:	7
All references:	8

Introduction

After the point where I integrated JPA relation mapping to Henaknowledge. I definitely I needed a way to first authenticate users and then based on their role they have student, teacher or admin they are authorized.

However, integrating such feature in spring boot is totally new for me and thus I surf the web on what authentication is and how it can be integrated in spring boot. After that I looked into what authorization is and how it can be integrated in spring boot as well.

- DOT framework methods used:

- [Available product analysis](#)
- [Prototyping](#)
- [System test](#)

Research question:

- Main question: What is a library or extension (that preferably built-in) that provide authentication and authorization features for spring boot project and what does it mean for the client side?
- Sub questions :What are authentication and authorization ? What is required for them to work ? How can they be integrated in spring boot?

The required criteria for completion

- The library or the extension has to provide authentication and authorization features in a secure way.
- The library or the extension preferably built-in
- Library or the extension should be free if not a built-in extension

Results of the research:

- In this research I used multiple resources. First resource is <https://www.youtube.com/> the reason being that it has some useful videos that can display existing products or features and explanation about them. Second resource is some other websites that explain much more in details.

Step 1:

- I started things off by trying to understand what authentication is and what are the requirement for it to work. Therefore, I checked the links provided in canvas and I read this article (Java dev, 2020) then the definition of authentication and authorization became clear to me. Authentication is to check the credentials provided by the user in order to let the user login in an application. Authorization is the second part which is all about privileges a user has. The limitation is that requires the principle of having different roles to exist for authorization to work. Thus, if an application does Not have based role principle then it only needs authentication and not authorization.

- After I read the article I decided to implement the tutorial it has. I created a new testing project to follow the tutorial and I successfully implemented it. Some of the issues I faced were that some existing packages like JWT were outdated and thus I look on the Github about the package was used in this tutorial and I managed to change it to the most updated one to fix the problem. Link: <https://github.com/auth0/java-jwt> (to be more specific, the version used in the tutorial was 3.11. However, I used 3.16)

- DOT framework methods used:
- [Available product analysis](#)

Step 2:

- Now that I finished with the first part of understanding the definition of what authorization and authorization are and got my hand dirty with a tutorial about them. I was kind of confident more about them. The next step I took was to look on existing YouTube videos regarding them since I am more comfortable on hearing a person explaining rather than reading and to get deeper and deeper in the subject.
 - Here are the links of the YouTube videos I watched: I highly recommend going in the same order I will mention the links. First this <https://www.youtube.com/watch?v=iyXne7dIn7U> then <https://www.youtube.com/watch?v=Tnt3GHuayXs> and lastly <https://www.youtube.com/watch?v=X80nJ5T7YpE>
 - Now that I got deeper and deeper into authentication and authorization and understood how to integrate them with spring boot as explained in the videos above and did some practical regarding them I was confident to start integrating authentication and authorization in Henaknowledge. However, all the information I gathered were more less like pieces and I had to put everything in the picture. For example, set up specific authorities for Henaknowledge differs from the videos or the tutorials I followed. Thus I changed the code to work with the authorities I set (Example, admin can have full privileges, unlike teacher or student that are limited.)
- DOT framework methods used:
 - [Available product analysis](#)
 - [Prototyping](#)

Step 3:

- At this stage, everything was working in the backend, every time I sent an object that consists of a username and a password in a post method to specific URL, I get token.

Here is one of the existing tokens my backend produced:

JWT:

```
eyJhbGciOiJIUzI1NiJ9.eyJyb2xlIjojU3R1ZGVudCIsInN1Yil6ImJiliwiaWF0IjoxNjIxNDA2MjEyLCJleHAiOjE2MjIyNzAyMTJ9.mZW6DxeVi2G6HOzNWCnhn0NhdOT6mfbTloKLeTb4y-4
```

- If you open the following website link, you can paste the token : <https://jwt.io/>

And this is the results you will get

PAYLOAD {

```
"role": "Student",
"sub": "bb",
"iat": 1621406212,
"exp": 1622270212
}
```

- All I had to do at this point was to decode the json web token generated by the backend in the frontend which is ReactJS.
- Therefore I used a different information source for looking into existing project or packages. Here is a link for the new source I used <https://www.npmjs.com/>
- I found this package that is capable of decoding the json web token and then I used the results to authorize the authenticated user. <https://www.npmjs.com/package/jwt-decode>

- DOT framework methods used:

- [Available product analysis](#)
- [Prototyping](#)
- [System test](#)

Conclusion:

The extension was a built-in interface that provide authentication an authorization services in spring boot projects that which most of the time lies in configuration part of the application. Next, I managed to understand the definition of authentication and authorization. Where they are performed and how can they be integrated with one of the micro servers (Spring boot). I managed to implement them in Henaknowledge project and I looked into how to extract the information hashed into the token the frontend receives and managed to implement that as well. Overall, the research results were very satisfying for me as a full stack developer.

All references:

A. (2021). *Auth0/java-jwt*. GitHub. <https://github.com/auth0/java-jwt>

auth0.com. (2021). *JWT.IO*. JSON Web Tokens - Jwt.Io. <https://jwt.io/>

Dev, J. (2020a, December 21). *Guide to use Lombok In Spring Boot*. Java to Dev.

<https://javatodev.com/lombok-spring-boot/>

Dev, J. (2020b, December 21). *Spring boot JWT authentication using spring security*. Java to Dev.

<https://javatodev.com/spring-boot-jwt-authentication/>

Dev, J. (2020c, December 21). *Spring Boot REST API Using JPA, Hibernate, MySQL Tutorial*. Java to Dev.

<https://javatodev.com/spring-boot-mysql/>

Java Brains. (2019a, August 17). *How to configure Spring Security Authentication - Java Brains*. YouTube.

<https://www.youtube.com/watch?v=iyXne7dIn7U>

Java Brains. (2019b, September 14). *Spring Boot + Spring Security with JPA authentication and MySQL from scratch - Java Brains*. YouTube.

<https://www.youtube.com/watch?v=TNt3GHuayXs>

Java Brains. (2019c, October 26). *Spring Boot + Spring Security + JWT from scratch - Java Brains*. YouTube.

<https://www.youtube.com/watch?v=X80nJ5T7YpE>

npm. (n.d.). Npm. <https://www.npmjs.com/>

npm: *jwt-decode*. (2020, November 16). Npm. <https://www.npmjs.com/package/jwt-decode>