**7. Spring Security**

**Reference :**

**https://www.bezkoder.com/spring-boot-jwt-auth-mongodb/**

1. Create a spring boot application with mongodb connection
2. Create a controller class and add 3 endpoints

* *“/home” – return a string “this is home page”*
* *“/dashboard” - return a string “this is dashboard page”*
* *“/manage” - return a string “this is manage page”*
* *“/login – validate login and issue token*

1. Add spring security config file with jwt authentication
2. *“/home”* should be accessible for any user (no credentials needed)
3. In order to access *“/dashboard”* or *“/manage”* users should provide valid user credentials (username and password).
4. *“/dashboard”* should be only accessible for users with role ADMIN or USER
5. *“/manage”* should be only accessible for users with role ADMIN
6. Test the API with postman and attach screenshots for the result of these scenarios
   1. call /home endpoint (without token)

Graphical user interface, text, application, email

Description automatically generated

* 1. call /dashboard and /manage endpoints without token

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* 1. login with invalid username

Graphical user interface, text, application, email

Description automatically generated

* 1. login with valid username but invalid password

Graphical user interface, text, application, email

Description automatically generated

* 1. login with user with USER role

Graphical user interface, text, application, email

Description automatically generated

* 1. call /dashboard and /manage endpoints with received token

Graphical user interface, text, application

Description automatically generated

* 1. login with user with ADMIN role

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* 1. call /dashboard and /manage endpoints with received token

Graphical user interface, text, application, email

Description automatically generated

1. Create new branch “spring-security” and push the project you created
2. Add your codes and answer sheet to a directory named “spring-security-training” and push it to your training github repository
3. Create a pull request to main branch and assign it to your trainer

**Additional:**

JWT stands for JSON Web Token, it is a compact, URL-safe means of representing claims to be transferred between two parties. A JWT token is used for securely transmitting information between parties as a JSON object. It is often used as a means of authentication and authorization in RESTful APIs and other web-based applications. JWT tokens are signed either using a secret or a public/private key pair, which enables the recipient of the token to verify its authenticity and ensure that it has not been tampered with.