Web Applications 2

Lab 4: Securing the application

Learning Objectives

By successfully completing this lab, students will develop the following skills:

- Configure the Spring Security library and implement role based access control
- Deploy the first microservice-based application consisting of two components: the access controller and the API implementation
- Using docker to manage component deployment

Description

Authentication is a critical component of any system that requires user identification and access control. In a ticketing system, users need to log in to create, manage or solve tickets. However, managing user authentication and authorization can be complex and time-consuming, especially when dealing with multiple applications or services.

Keycloak is an open-source identity and access management solution that simplifies user management for modern applications and services. By integrating with Keycloak, we can leverage its features and benefits, such as centralized user authentication, single sign-on, and fine-grained authorization.

In this laboratory, we will create an authentication server that communicates with Keycloak to handle user authentication and generate JWTs for authorized users.

We will add the authorization part to the ticketing service, developed in the previous lab, so that with the role contained in the jwt you can authorize or not the various apis

Steps

1. Starting from lab 3 repository, a member of the team extends the solution adding as dependencies Spring Security and Oauth2 Resource Server. Commit and push
2. Configure and deploy a Keycloak docker container following the documentation https://www.keycloak.org/
3. Navigate to Keycloak address and create a new Realm, Client a set of roles (Expert, Manager, Client). For this lab users must be created manually using the Keycloak GUI
4. Implement the login flow: in the authentication server project, create a new endpoint for handling login requests. When a user submits their credentials, the server should validate them by sending a request to the Keycloak server. If the credentials are valid, Keycloak will return a JWT that the server can use to authorize the user for

subsequent requests. The JWT will be returned to the client that will use it in subsequent requests.
https://medium.com/geekculture/using-keycloak-with-spring-boot-3-0-376fa9f60e0b

5. Add the Authorization layer to the ticketing service by providing a suitable @Configuration class and modify existing endpoints/service methods so that only those that are allowed according to the business rules can access them.
6. Create a docker image of the new microservice with jib, and run it locally.

Submission rules

Download a copy of the zipped repository and upload it in the “Elaborati” section of “Portale della didattica”. Label your file “Lab4-Group<N>.zip” (one copy, only - not one per each team member).
Work is due by Friday May 19, 23.59 for odd numbered teams, and by Friday May 26, 23.59 for even numbered ones.