Task 3. Authentication and authorization for Developer API

We will continue to improve our Developer API from the previous task. Let's add authentication and authorization using JWT (JSON Web Tokens) to our API.

First, add one endpoint to our API. It should return the generated JWT token. The endpoint contract is as follows:

@RequestMapping(value = "/builder-jwt", method = POST)
public JwtResponse builderJWT(@RequestBody Map<String, Object> claims).....

Generated token must contain the following claims: issuer, subject, expiration date, roles. You can add others of your choice. Roles is an array of strings from the enum:
ADMIN, USER, HR.
JWT must be signed with a string SecretKey. SecretKey should not be hard-coded, but should be passed to the application via an environment variable:

SECRET_KEY=”12345joipuiu98uoiupoiuhpuhhoiuy9uuhi98hliugfkjgliulkjlkjhliuhlkhlkj”

Request example:

POST /builder-jwt HTTP/1.1
{
"iss": "GP",
"sub": "task2",
……
“roles” : [“ADMIN”, “HR”]
}

response example with JWT:
"eyJhbGciOiJIUzUxMiJ9.eyJzY29wZSI6ImRldiIsInJvbGUiOlsiZmlzIl0sImlzcyI6IlNWUGEyZyIsImlhdCI6MTY0NDQ0NTEyOCwiZXhwIjoxNjQ0NTMxNTI4LCJqdGkiOiI0ZmNiYTU5OC1iYWY3LTQ3ODktOGU0Yi0xZjM0MzBmOTQ5ZmUifQ.R4YepITz-SexRSLehOQLZQTRXrd5Ua4mdjWpgu-8W2MPtTq0kWBTWFOe5JqCpZufB1GoPKrZ-4VXOqvExe9D0A"

Please, add this new endpoint to your Swagger for Developer API. Your Swagger UI should be runnable locally and we will try to check it. Describe the details in README if necessary.

Hint: you should use JJWT (https://github.com/jwtk/jjwt) is a Java library providing end-to-end JSON Web Token creation and verification. There are also online parsers for JWT (https://jwt.io).

Second, we allow only authenticated users with role ADMIN to perform create, update, delete and read operations for our Developer API (GET, POST, PUT, DELETE

methods). Users with roles USER and HR can only perform read-only operations for our API (GET method).

How should it work?

Requests to API include an "Authorization" header with the value "Bearer OUR_JWT_TOKEN". If there is no token, then the user is considered unauthenticated and receives a http-response 401 Unauthorized.

If there is a token, then the application validates it (for example, using library JJWT) and checks claims.

If a user with the roles of USER or HR tries to execute POST, PUT or DELETE methods, then he receives a http-response 403 Forbidden. If the user does not have any roles in JWT, then he receives a http-response 403 Forbidden.

If any user has the token's 'issuer' != "GP", then he receives http-response 403 Forbidden.

Hint: you can use the Spring Security module.

We will check the operation of the application as follows:

- using your Swagger to generate several tokens with different roles and without roles,
- make requests to the API with these tokens and without a token,
- make sure the application works as described above.

If something was not specified in the requirements, then you can check them with us, or do it at your discretion. No need for new Unit tests. Please, use Spring Boot, Spring JPA, in memory database (h2), lombok, Java 11+.

We'll be evaluating your submission from the following perspectives:
* Code quality and best practices for new code changes.
* Implementation of new features

This is how we are going to run and evaluate your submission, so please make sure to run below steps before submitting your answer.

- Make sure the application is compiling, running and all dependencies are included.
- Commit everything to your github repo
- Send a link of github to the GP team.