OAuth2 + JWT hands on

1. We added a starter dependency “spring security”

After adding this dependency, when we run the project,

A password was generated and displayed in console.

When we try to access any url in our webservice, we first found a login page automatically coming.

How did we login?

Username: user

Password: we copied that generated password from console.

1. We wanted to decide our own password. Also our own username.

We configured this in our application.properties

spring.security.username=jag

spring.security.password=jag@123

1. We did not want to configure the user credentials in text file. We wanted to do it in program.

UserDetailsService

Configure method returns

new User(“jag”,”jag@123”,new ArrayList<>());

1. We did not want dummy user credentials. Instead we want to take it from database. An usermaster table can contain several rows, we find a row by username. And return that row as an object of user.

We used JPA repository

1. JWT

In restful webservices, every method call is independent. STATELESS

1. Oauth2

Instead of gmail or facebook or github, we can create our own clientid and clientsecret using open ssl tool.

Task:

Kindly do the above 1) , 2) and 3)

If we create restful web services, 3 web services, and we test them in postman.

How will a web page come and ask for username and password?

Not possible, so we will create our own authenticate method (postmapping) and if login is successful, then we respond with a JWT token

JWT - JSON Web Token

Dependency:

Jjwt

1. JwtUtil

Create token

Generate token

extractUserNameFromToken

1. JwtFilter

Which can intercept all requests coming to DispatcherServlet and can read the header (authorization header) and check the jwt and check if it is valid or expired and can extract username, etc.

1. Configure

So that few urls are permitAll and other urls are authenticated

1. AuthRequest.java

username

password

getters and setters

why? For login, we use post mapping, request body authrequest object

1. After login successful, respond with an object

JwtToken

String jwt

Demo:

1. Create a spring boot project “20-jul-security-jwt”
2. Dependencies:

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt</artifactId>

<version>0.9.1</version>

</dependency>

1. Kindly refer to the project

\advanced\20-jul-security-jwt

We have completed signup and login.

Now, when login is successful, we should get a token. JWT.

1. Dependency is already added
2. JwtUtil.java
3. JwtFilter.java
4. MySecurityConfiguration.java

If login is successful, instead of returning a String “success”, we are going to return an object that contains jwt token

1) signup POST

localhost:8082/ust/signup

{"username":"jag","password":"jag"}

2) login POST

localhost:8082/ust/login

get the token now

3) GET header: Authorization Bearer <<token>>

localhost:8082/ust

welcome to home

3) GET use the authorization header with token:

localhost:8082/contact

Contact us

Same project : \advanced\20-jul-security-jwt

Git branch renamed now.

Git branch removed now.