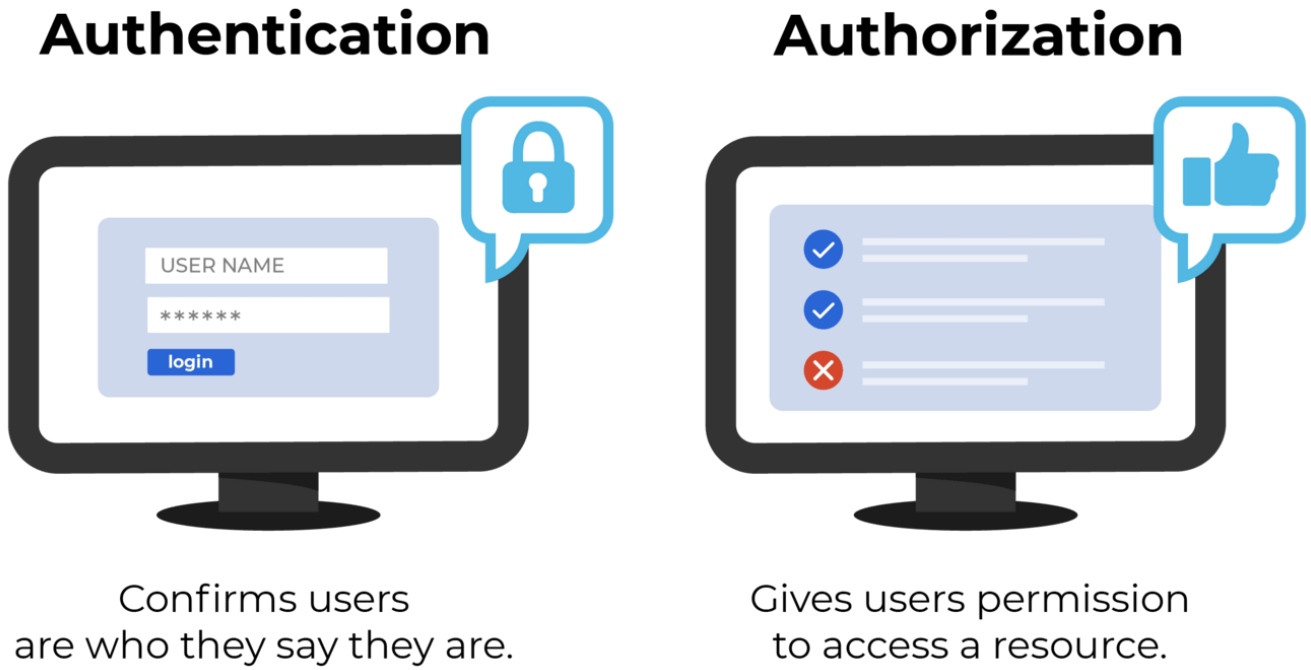
Spring Security + JWT

Prerequisites

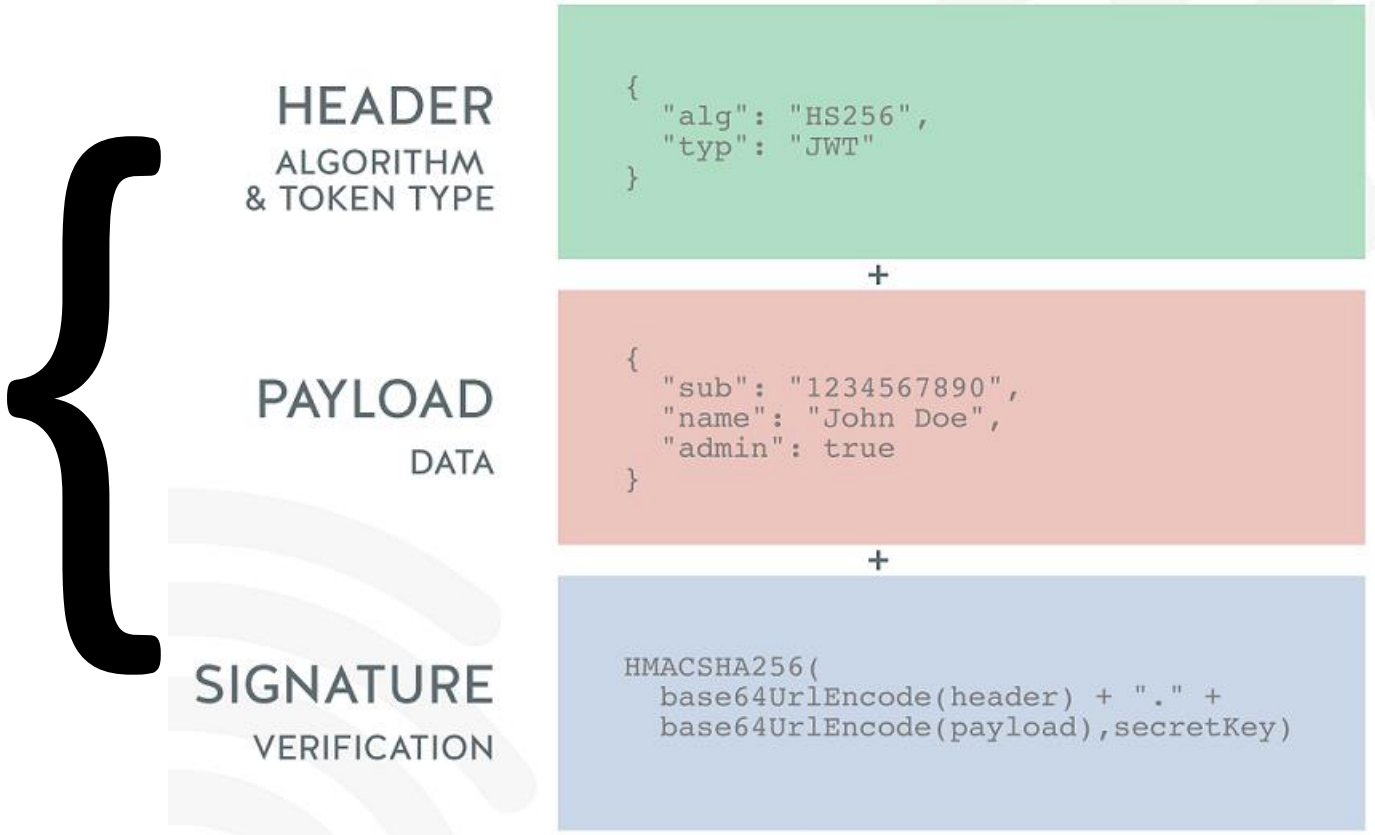
- Spring boot
- DataJPA

Authorization & Authentication



JSON Web Token is an open standard (RFC 7519) that defines a compact and autonomous way to securely transfer information between parties as a JSON object.

Components Of JWT



The header consists of 2 fields. The typ whose value is JWT and the hashing algorithm to be used.

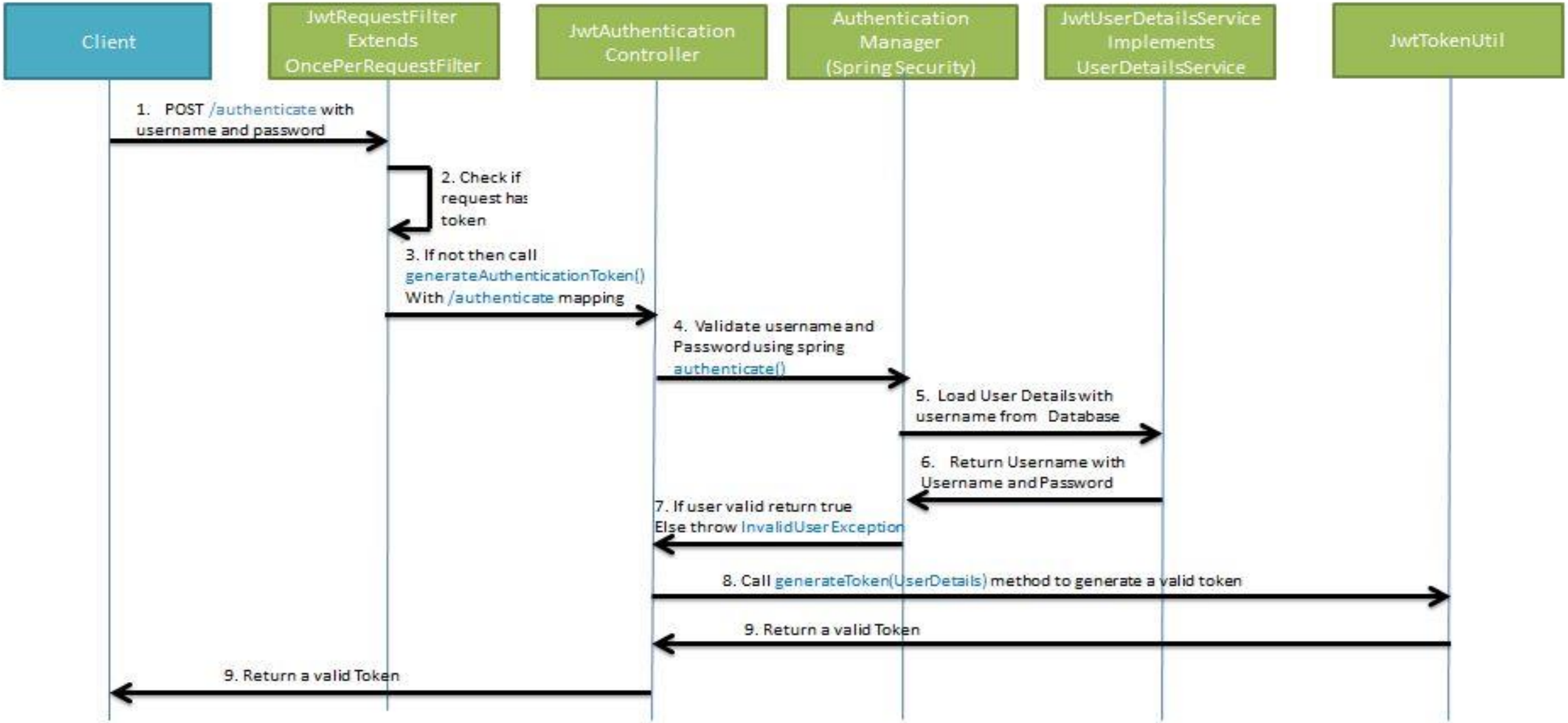
Payload consists of data to be exchanged securely This information represents claims. The Claim Names within this object MUST be unique

Signature is the hash of the header and the payload using the hashing algorithm specified in the the Header and the secret which is known only to the server

eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyMTIiLCJleHAiOj
E2NDI2NjQ3NTksImlhdCI6MTY0MjY2Mjk1OX0.pjH9PjCC
DUh6JtLbj4EbRIbC2Fhxb4SqJMTVsGKHy5s

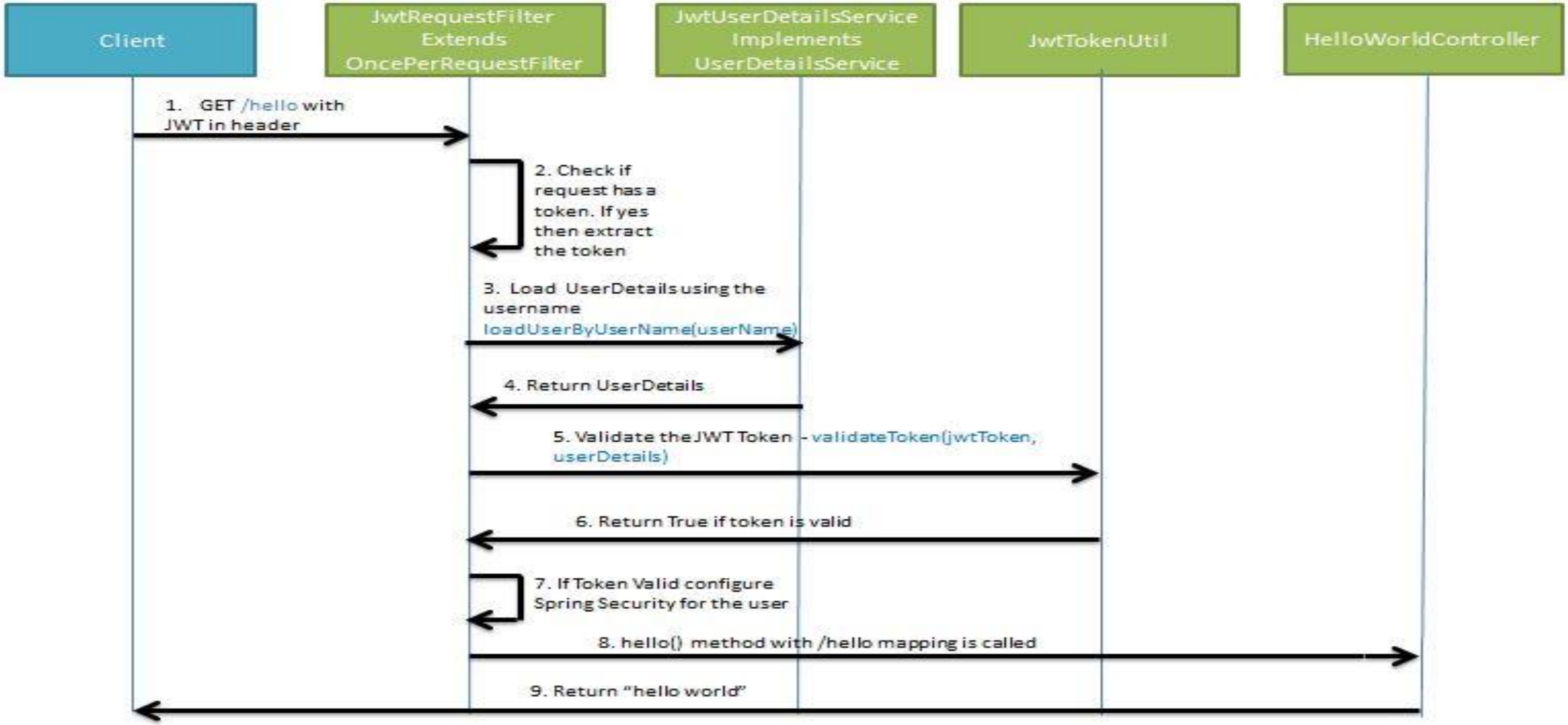
Sample JWT

Generate Token




Username and Password provided by the User
AuthenticationManager
Username and Password from the database(provided by the JwtUserDetailsService)
If Authentication Successful return true else throw invalid user exception

Validate Token



Spring Security and JWT dependencies

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
        <groupId>io.jsonwebtoken</groupId>
        <artifactId>jjwt</artifactId>
        <version>0.9.1</version>
</dependency>
```

Setting up the application.properties

- The secret key is combined with the header and the payload to create a unique hash. We are only able to verify this hash if you have the secret key.
  - jwt.secret=java

Setting up the Token utility →JWtTokenUtil

- The JwtTokenUtil is responsible for performing JWT operations like creation and validation.
- It makes use of the io.jsonwebtoken.Jwts for achieving this.

Setting up the userDetails for JWT access

- Can be → JWTUserDetailsService/UserDetailsSeriveImple
- JWTUserDetailsService/UserDetailsSeriveImple implements the Spring Security UserDetailsService interface.
- It overrides the loadUserByUsername for fetching user details from the database using the username.
- The Spring Security Authentication Manager calls this method for getting the user details from the database when authenticating the user details provided by the user.