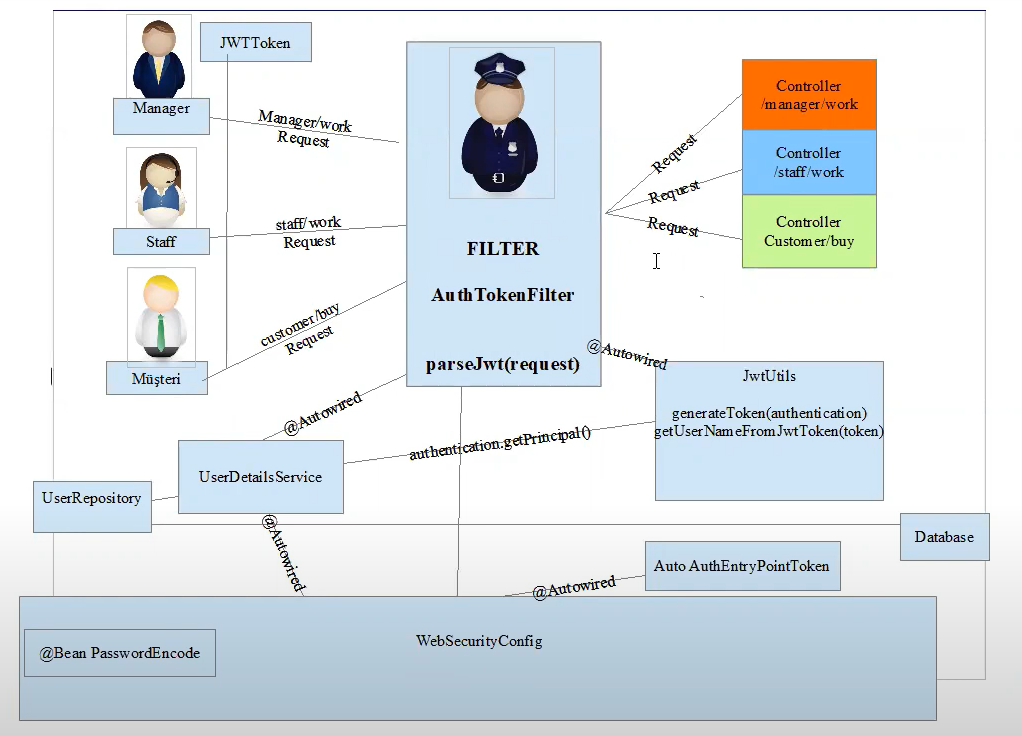
SPRING SECURITY “LOGIN” AND “REGISTER”



Register and Login methods do not need permission but other requests go to the filter.

\*How to register

1.The client calls register method. Then client comes with its own information in “**public class RegisterRequest**”

2. With registerUser() method in “**public class UserJWTController**” we call registerUser() in “public class UserService”

3.

a)In “public class userService” by registerUser() method first of all we check if unique fields(ssn,email and username) already exists and type code for exceptions.

b)we create a default role(customer) for the user and we add this role to the set named “userRoles”

c)By using constructor we create a user

d)By using userRepository object we save user.

After saving the user we need to create an account for this user. This is done by using **public class AccountService**

4.In **public class AccountService** by method createAccount() we set balance(default 0.00), account number and user

\*How to login

1.Client login request is accepted by the “login” method in **public class UserJWTController**

Then we call authenticate() method. This method accepts UserNamePasswordAuthenticationToken. We keep username and password in LoginDTO for this method.

This code goes on if authentication is successfull. After successful authentication token is generated. (authentication object instance ise used in generateToken() method.

2.generateToken() method is typed in **public class JwtUtils.** By this method we get the principals. THIS MEANS CURRENT USER. We cast this principals to UserDetailsImpl, a class which implements interface UserDetails:

/\* Here is details for interface UserDetails:

Provides core user information. Implementations are not used directly by Spring Security for security purposes. They simply store user information which is later encapsulated into [Authentication](eclipse-javadoc:%E2%98%82=surebank/C:%5C/Users%5C/Asim%5C/.m2%5C/repository%5C/org%5C/springframework%5C/security%5C/spring-security-core%5C/5.6.2%5C/spring-security-core-5.6.2.jar=/maven.pomderived=/true=/=/maven.pomderived=/true=/=/maven.groupId=/org.springframework.security=/=/maven.artifactId=/spring-security-core=/=/maven.version=/5.6.2=/=/maven.scope=/compile=/%3Corg.springframework.security.core.userdetails(UserDetails.class%E2%98%83UserDetails%E2%98%82Authentication) objects. This allows non-security related user information (such as email addresses, telephone numbers etc) to be stored in a convenient location. \*/

Then we take username from userDetails and issued date expiration date algorithm create a token and return a token into the “login” method at UserJWTController class. This token is used to create an instance of LoginResponse(a custom class created by us). Finally we return this response with static method ok of ResponseEntity class.