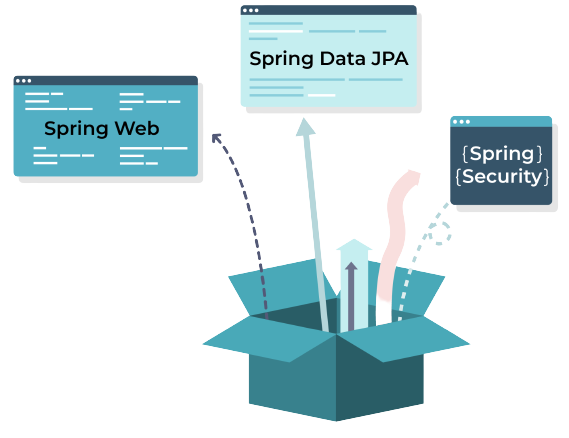
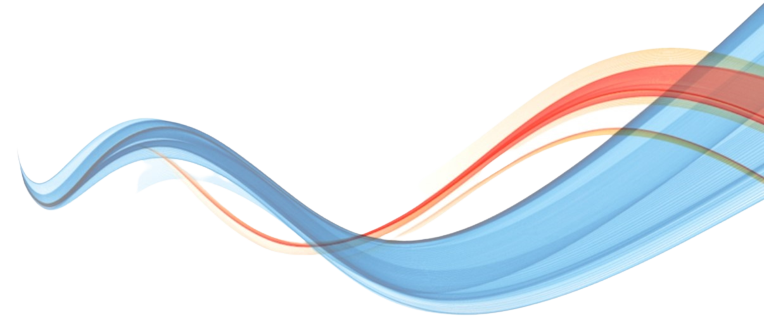
Compte rendu de :

SEANCE 05



**Réalisé Par :** AYOUJJIL Soukayna

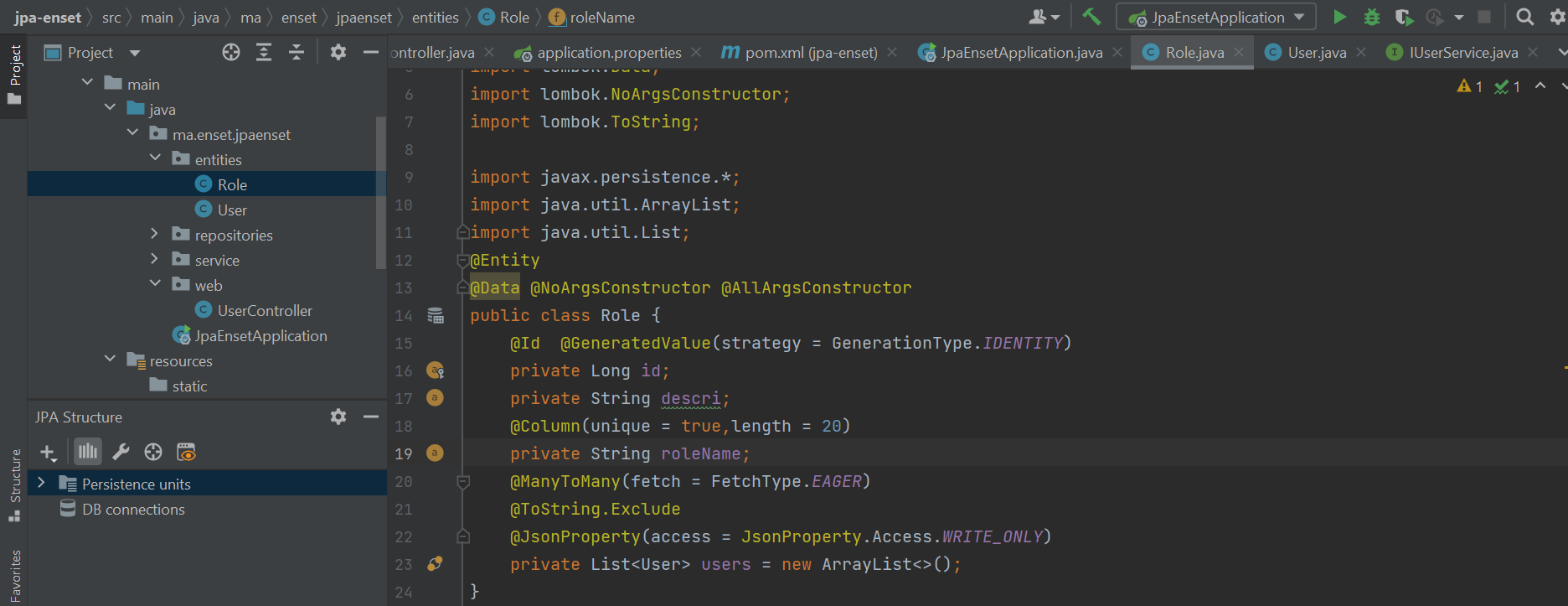
**Département  :** Mathématiques et Informatique

**Filière :** II-BDCC

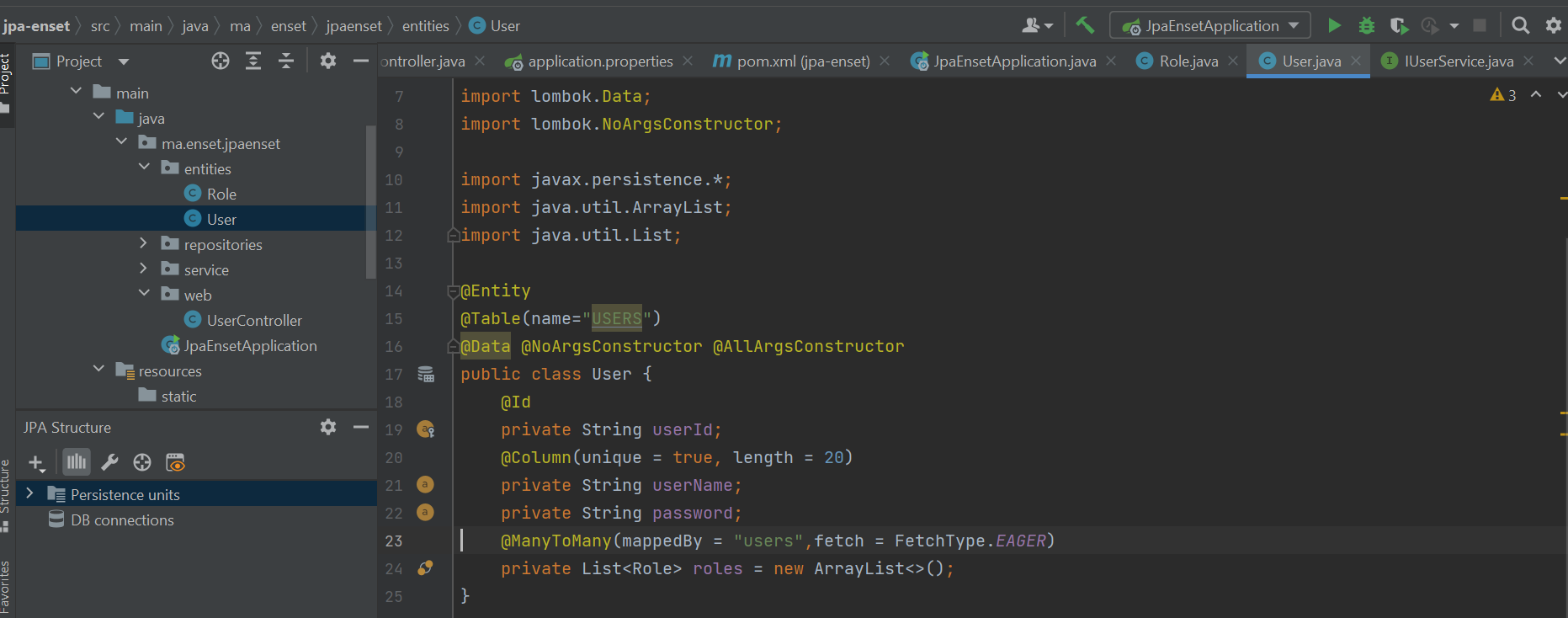
**Professeur :** M. Mohamed YOUSSFI

**Année Universitaire** **: 2021/2022**

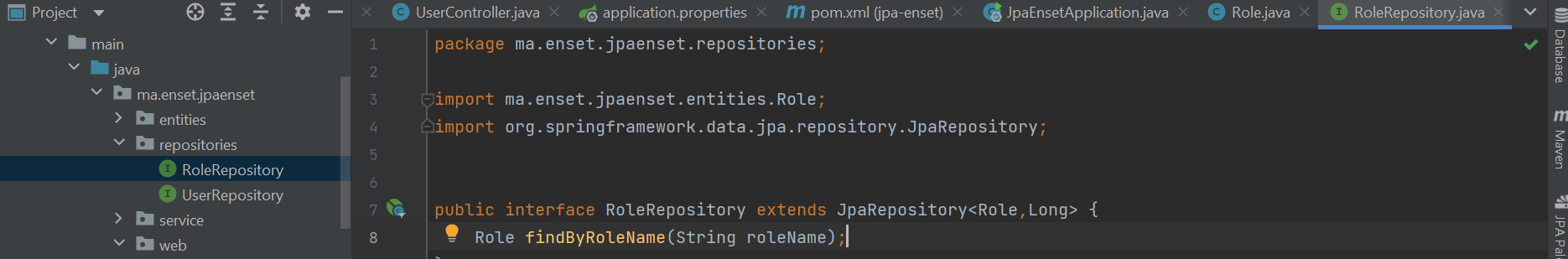
* Gestion des entités :
* Role



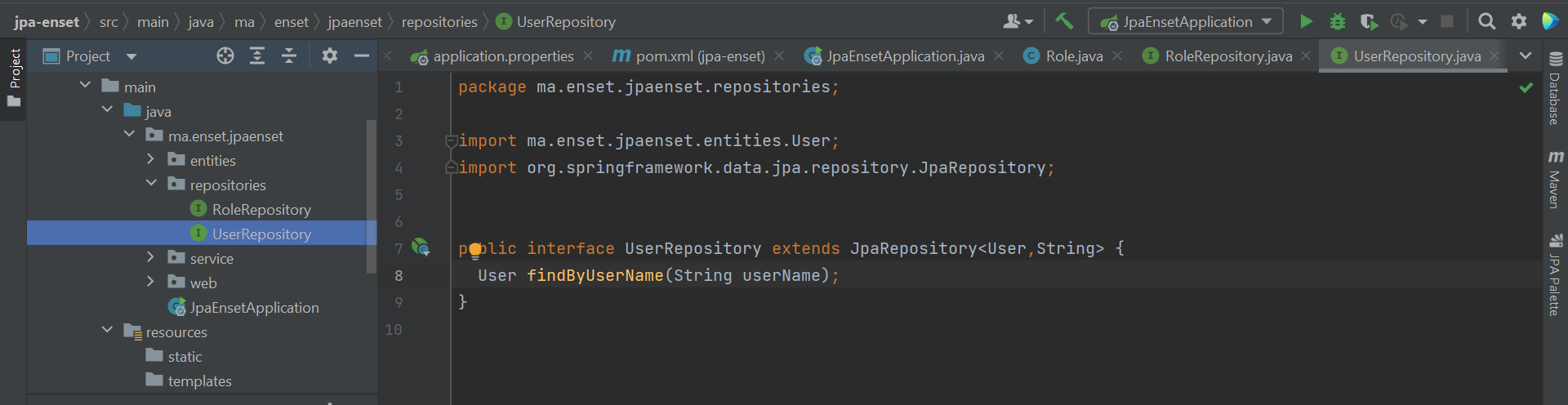
* User



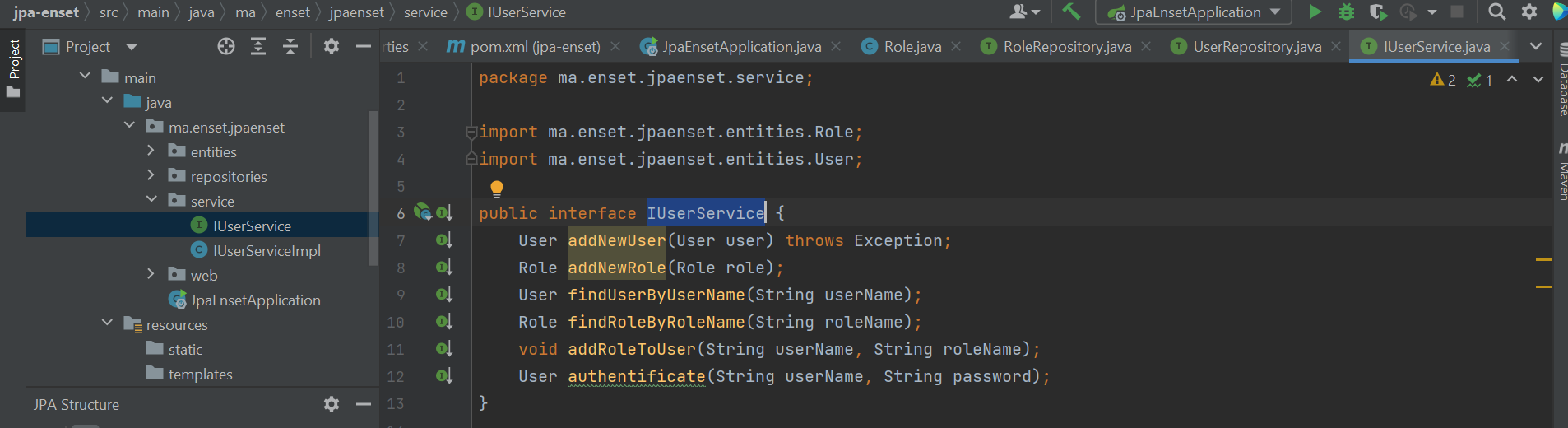
* Gestion des repositories
* RoleRespository



* UserRespository



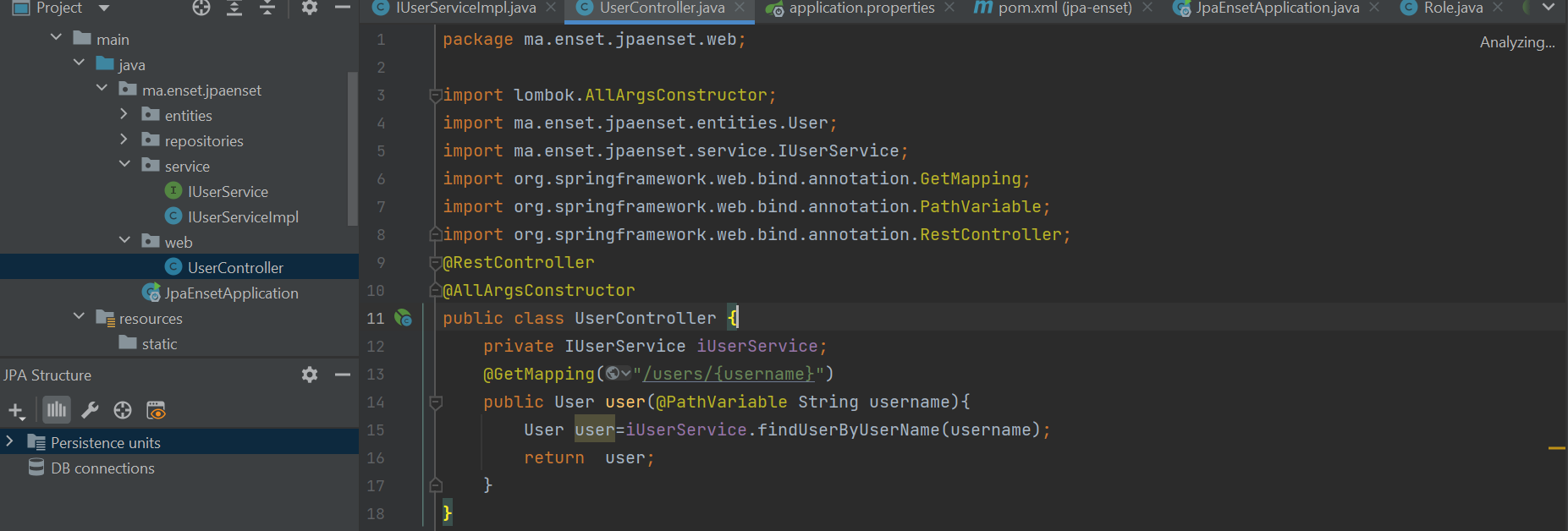
* Couche service
* L’interface IUserService



* L’implémentation de l’interface IUserService

import lombok.AllArgsConstructor;  
import ma.enset.jpaenset.entities.Role;  
import ma.enset.jpaenset.entities.User;  
import ma.enset.jpaenset.repositories.RoleRepository;  
import ma.enset.jpaenset.repositories.UserRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import javax.transaction.Transactional;  
import java.util.UUID;  
  
//service indique que la classe doit etre instancier au demarrage service , transactional  
@Service  
@Transactional  
@AllArgsConstructor  
public class IUserServiceImpl implements IUserService {  
  
 private UserRepository userRepository;  
 private RoleRepository roleRepository;  
  
  
 @Override  
 public User addNewUser(User user) {  
 user.setUserId(UUID.*randomUUID*().toString());  
 return userRepository.save(user);  
 }  
  
 @Override  
 public Role addNewRole(Role role) {  
 return roleRepository.save(role);  
 }  
  
 @Override  
 public User findUserByUserName(String userName) {  
 return userRepository.findByUserName(userName);  
 }  
  
 @Override  
 public Role findRoleByRoleName(String roleName) {  
 return roleRepository.findByRoleName(roleName);  
 }  
  
 @Override  
 public void addRoleToUser(String userName, String roleName) {  
 User user = findUserByUserName(userName);  
 Role role = findRoleByRoleName(roleName);  
 if(user.getRoles()!=null){  
 user.getRoles().add(role);  
 role.getUsers().add(user);  
 }  
 //userRepository.save(user);  
 }  
  
 @Override  
 public User authentificate(String userName, String password) {  
 User user = userRepository.findByUserName(userName);  
 if (user==null) throw new RuntimeException("Bad credentials");  
 if (user.getPassword().equals(password)){  
 return user;  
 }  
 throw new RuntimeException("Bad credentials");  
 }  
}

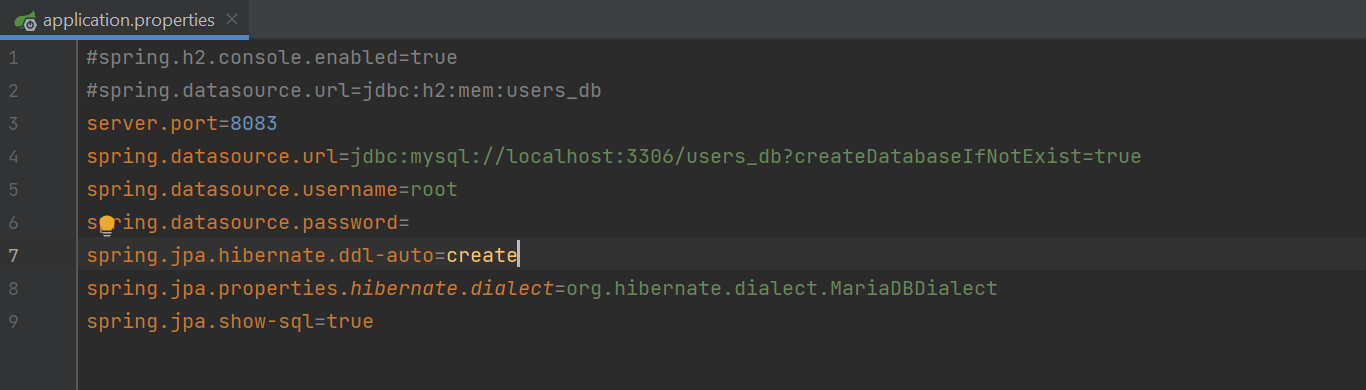
* Couche Web



* Création des users + affectation des rôles

package ma.enset.jpaenset;  
  
import ma.enset.jpaenset.entities.Role;  
import ma.enset.jpaenset.entities.User;  
import ma.enset.jpaenset.service.IUserService;  
import org.springframework.boot.CommandLineRunner;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.annotation.Bean;  
  
import java.util.stream.Stream;  
  
@SpringBootApplication  
public class JpaEnsetApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(JpaEnsetApplication.class, args);  
 }  
 @Bean  
 CommandLineRunner start(IUserService iUserService){  
 return args -> {  
  
 User user = new User();  
 user.setUserName("user1");  
 user.setPassword("user1");  
 iUserService.addNewUser(user);  
  
 User user1 = new User();  
 user1.setUserName("admin");  
 user1.setPassword("admin");  
 iUserService.addNewUser(user1);  
  
 Stream.*of*("STUDENT","USER","ADMIN").forEach(  
 r->{  
 Role role =new Role();  
 role.setRoleName(r);  
 role.setDescri("my role is : "+r);  
 iUserService.addNewRole(role);  
 }  
 );  
  
 iUserService.addRoleToUser("user1","STUDENT");  
 iUserService.addRoleToUser("user1","USER");  
 iUserService.addRoleToUser("admin","USER");  
 iUserService.addRoleToUser("admin","ADMIN");  
  
 try {  
 User user3 = iUserService.authentificate("user1","user1");  
 System.*out*.println(user.getUserId());  
 System.*out*.println(user.getUserName());  
 System.*out*.println("Liste des roles : ");  
 user3.getRoles().forEach(  
 r->{  
 System.*out*.println("Roles=>"+r);  
 }  
 );  
 }catch (Exception e){  
 e.getStackTrace();  
 }  
 };  
 }  
}

* application.properties



* Resultats :

