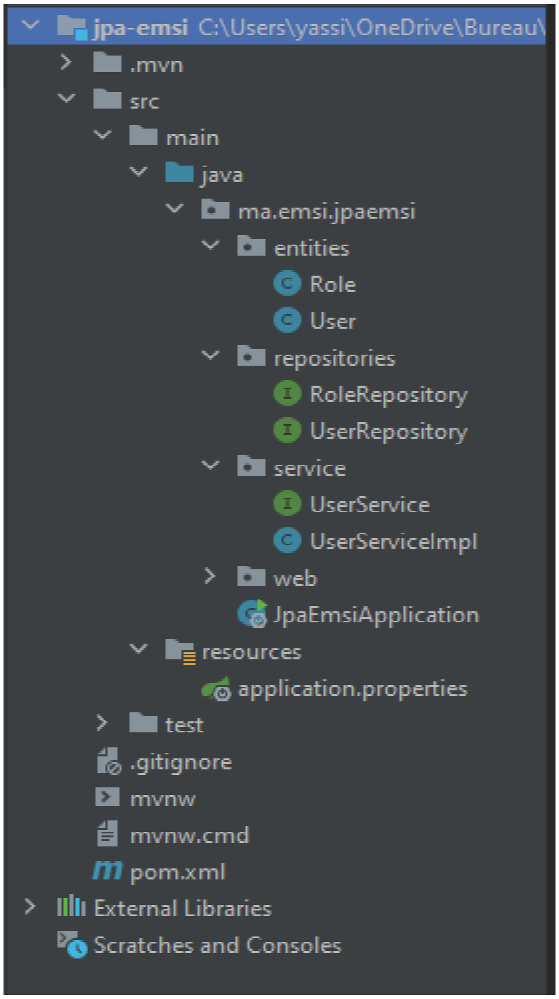
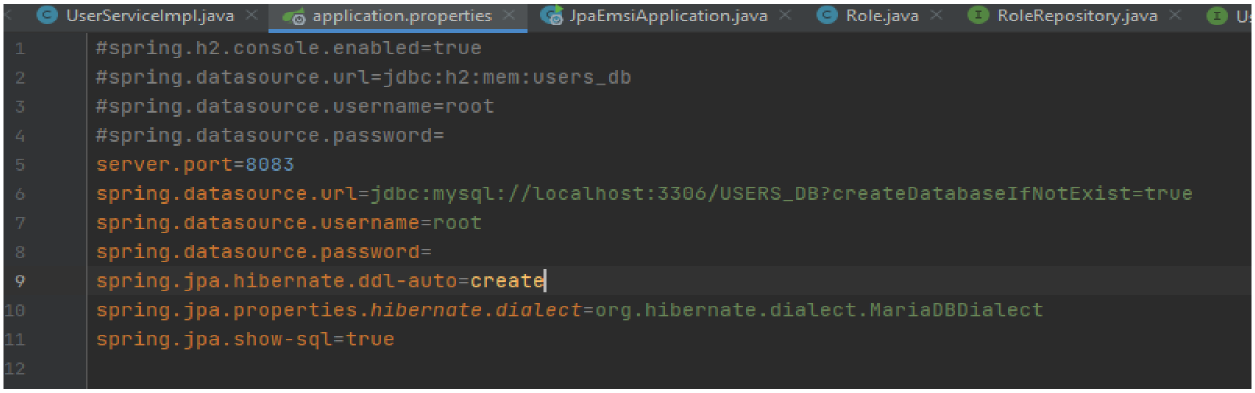
Acitivite 2 JEE/ Spring Boot - ORM avec Spring Data JPA

Hicham Yassine 4IIR G4

Partie 2 APP jpa-emsi

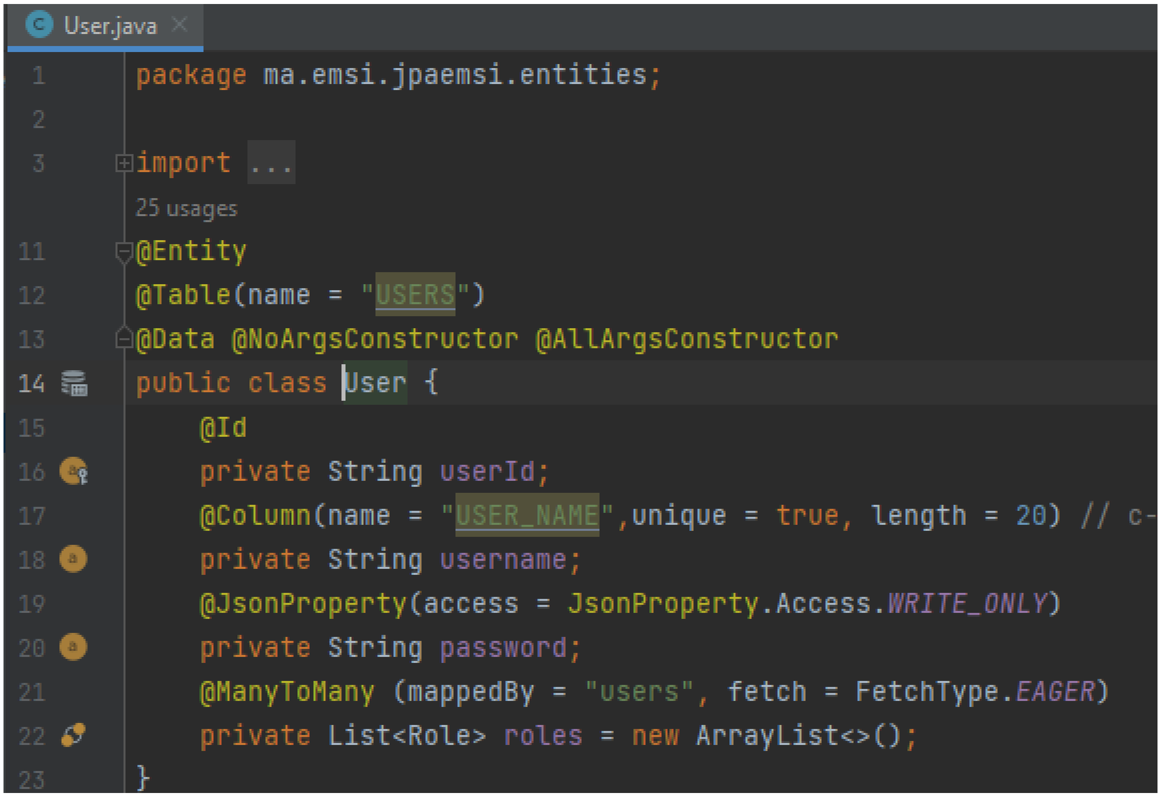


```
jpa-emsi C:\Users\yassi\OneDrive\Bureau\
  .mvn
  src
    main
      java
        ma.emsi.jpaemsi
          entities
            Role
            User
          repositories
            RoleRepository
            UserRepository
          service
            UserService
            UserServiceImpl
          web
          JpaEmsiApplication
      resources
        application.properties
    test
  .gitignore
  mvnw
  mvnw.cmd
  pom.xml
External Libraries
Scratches and Consoles
```

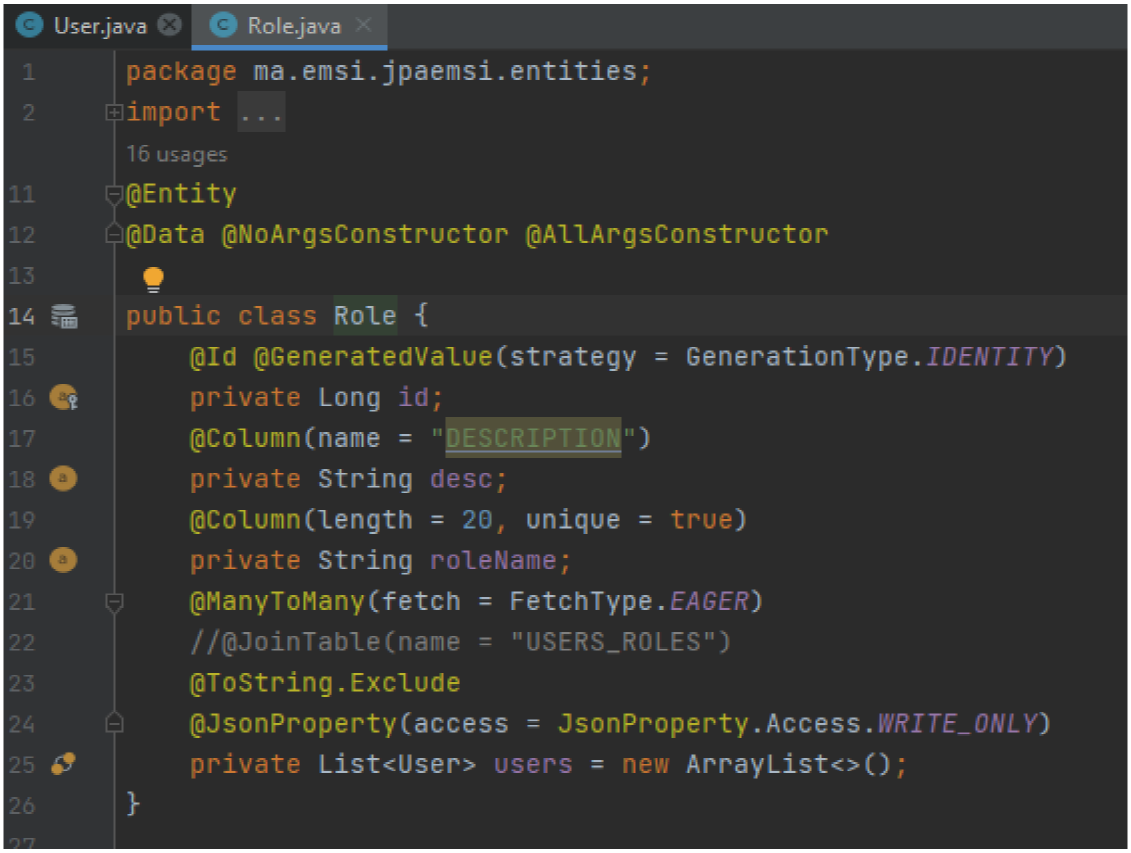


```
#spring.h2.console.enabled=true
#spring.datasource.url=jdbc:h2:mem:users_db
#spring.datasource.username=root
#spring.datasource.password=
server.port=8083
spring.datasource.url=jdbc:mysql://localhost:3306/USERS_DB?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=
spring.jpa.hibernate.ddl-auto=create
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MariaDBDialect
spring.jpa.show-sql=true

```

Les class

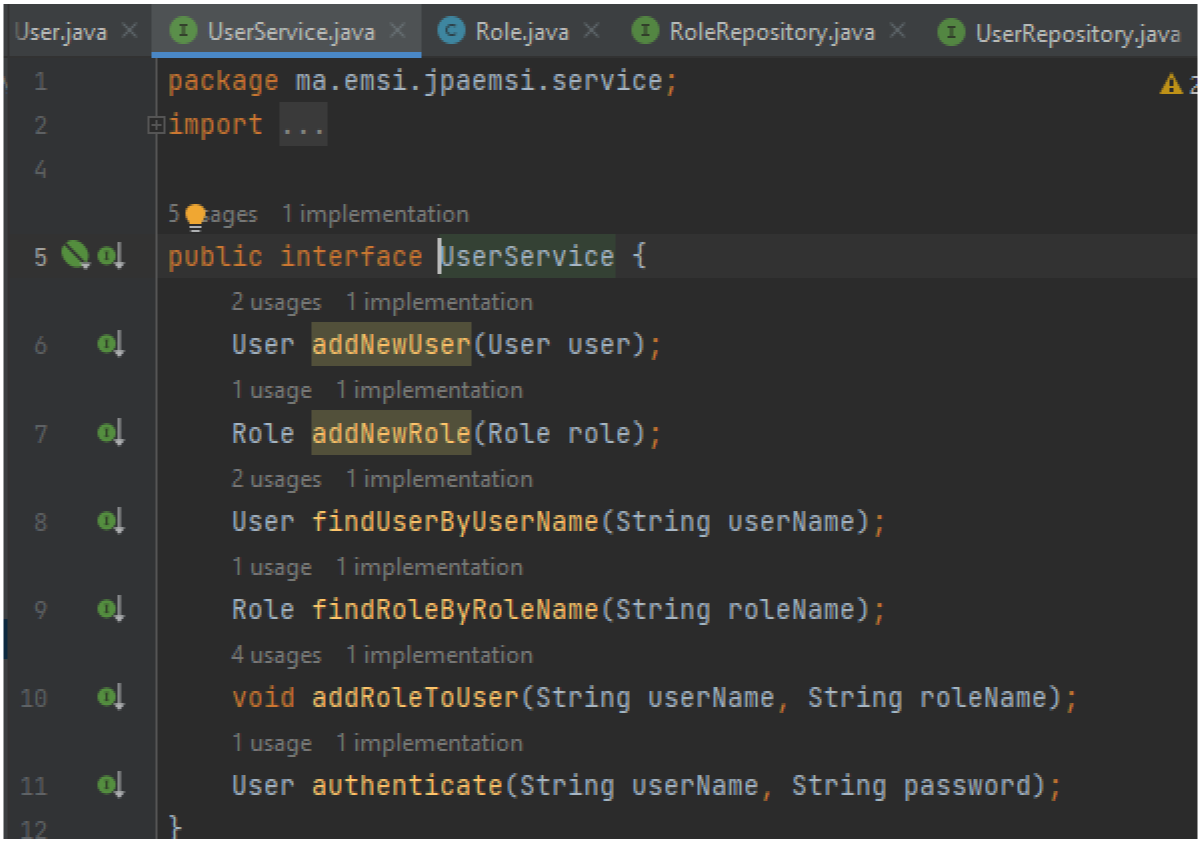


```java
package ma.emsi.jpaemsi.entities;

import ...
25 usages
@Entity
@Table(name = "USERS")
@Data @NoArgsConstructor @AllArgsConstructor
public class User {
    @Id
    private String userId;
    @Column(name = "USER_NAME",unique = true, length = 20) // c-
    private String username;
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private String password;
    @ManyToMany (mappedBy = "users", fetch = FetchType.EAGER)
    private List<Role> roles = new ArrayList<>();
}
```

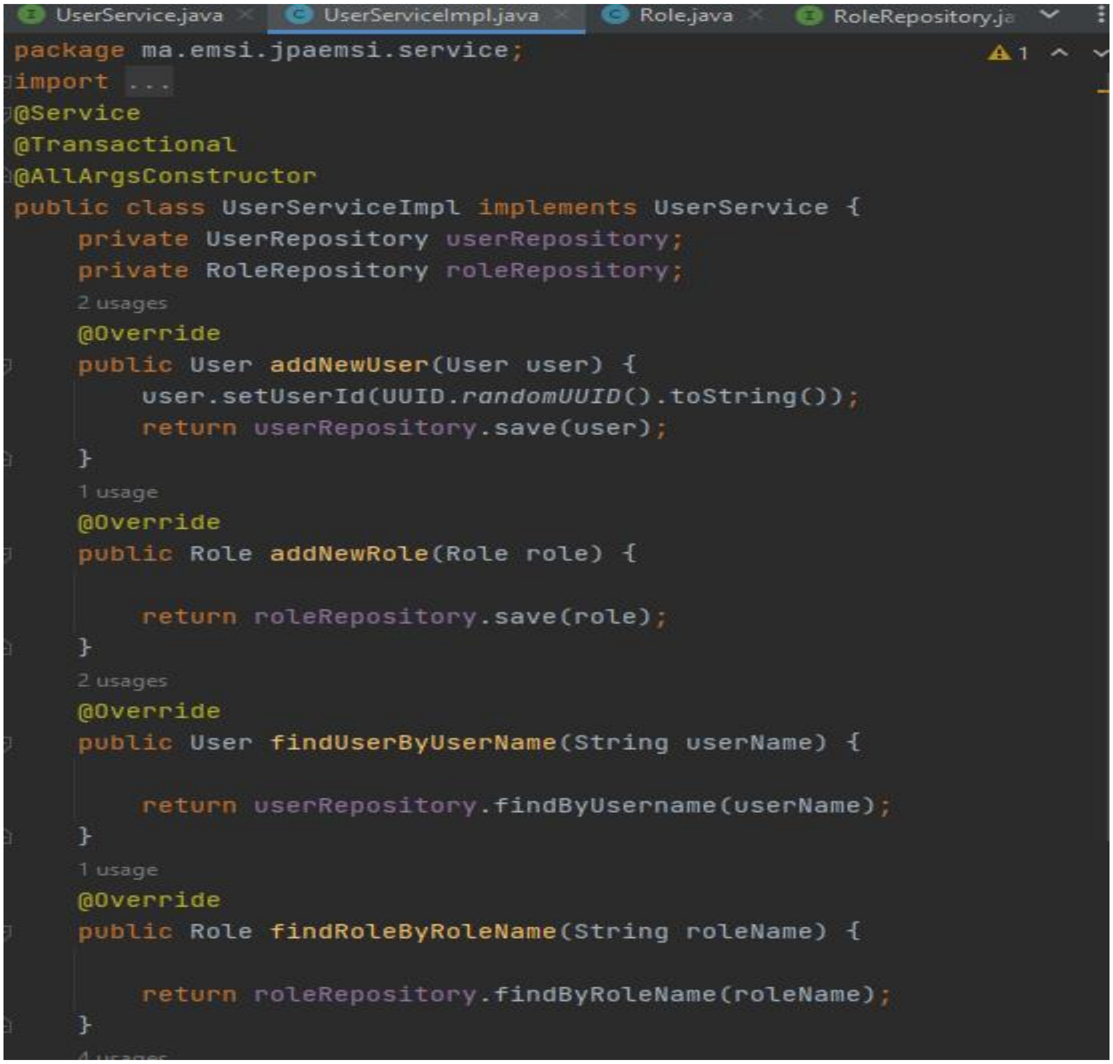


```java
package ma.emsi.jpaemsi.entities;
import ...
16 usages
@Entity
@Data @NoArgsConstructor @AllArgsConstructor

public class Role {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Column(name = "DESCRIPTION")
    private String desc;
    @Column(length = 20, unique = true)
    private String roleName;
    @ManyToMany(fetch = FetchType.EAGER)
    //@JoinTable(name = "USERS_ROLES")
    @ToString.Exclude
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private List<User> users = new ArrayList<>();
}
```

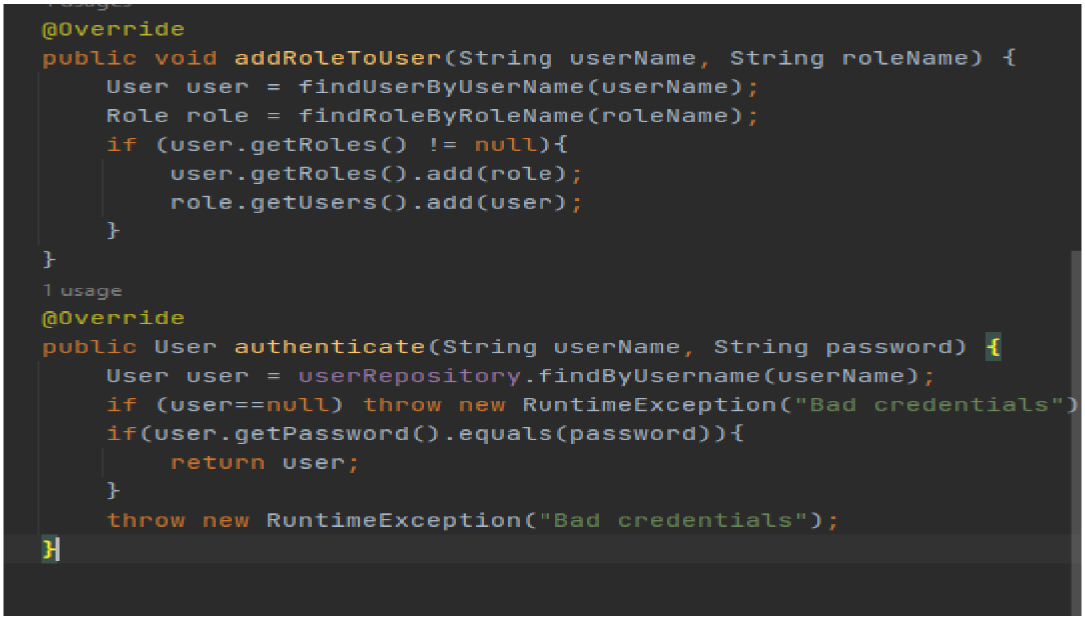
Service et son implemntation



```java
package ma.emsi.jpaemsi.service;
import ...

public interface UserService {
    User addNewUser(User user);
    Role addNewRole(Role role);
    User findUserByUserName(String userName);
    Role findRoleByRoleName(String roleName);
    void addRoleToUser(String userName, String roleName);
    User authenticate(String userName, String password);
}
```

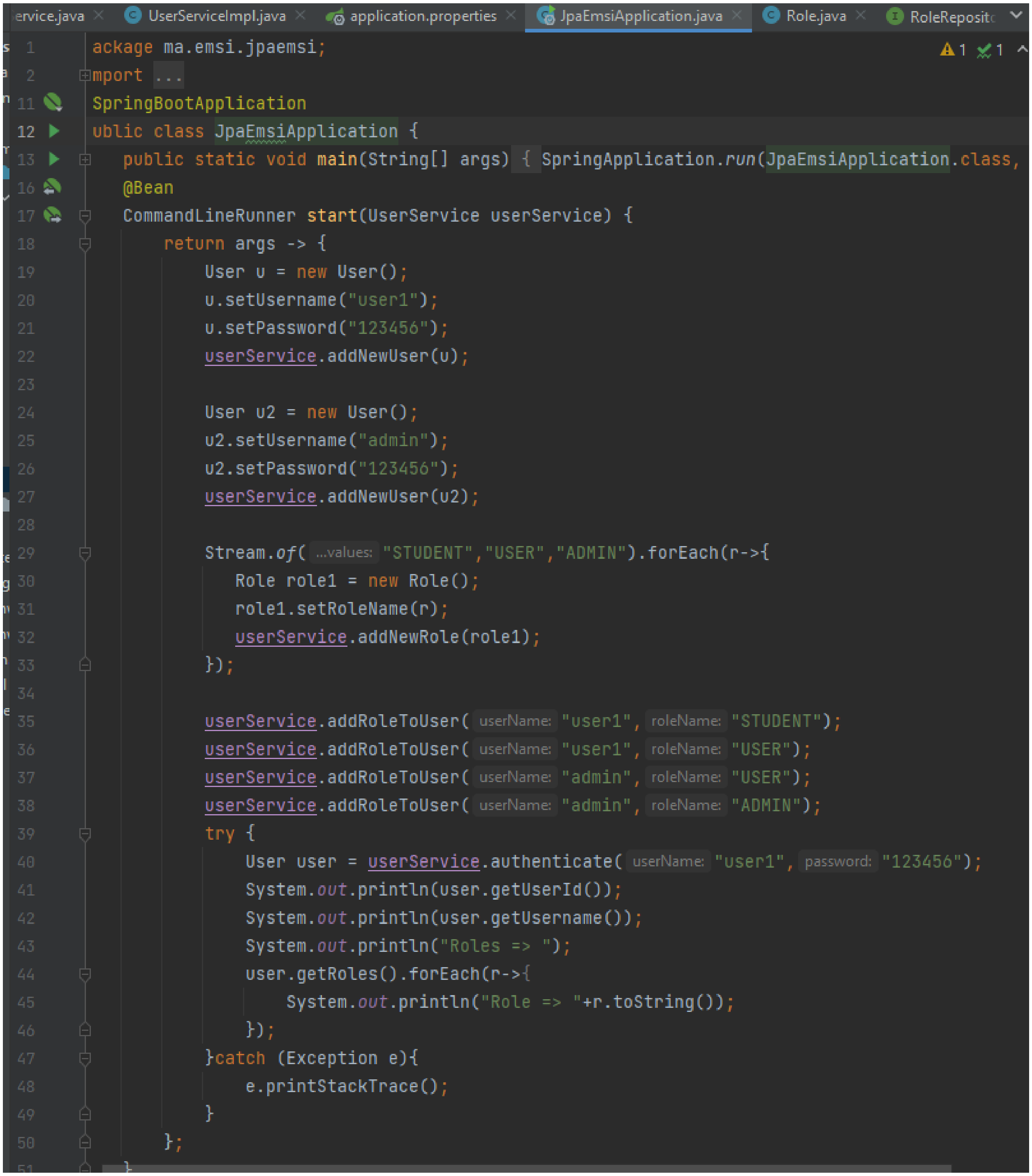


```java
package ma.emsi.jpaemsi.service;
import ...
@Service
@Transactional
@AllArgsConstructor
public class UserServiceImpl implements UserService {
    private UserRepository userRepository;
    private RoleRepository roleRepository;
    2 usages
    @Override
    public User addNewUser(User user) {
        user.setUserId(UUID.randomUUID().toString());
        return userRepository.save(user);
    }
    1 usage
    @Override
    public Role addNewRole(Role role) {

        return roleRepository.save(role);
    }
    2 usages
    @Override
    public User findUserByUserName(String userName) {

        return userRepository.findByUsername(userName);
    }
    1 usage
    @Override
    public Role findRoleByRoleName(String roleName) {

        return roleRepository.findByRoleName(roleName);
    }
    4 usages
```

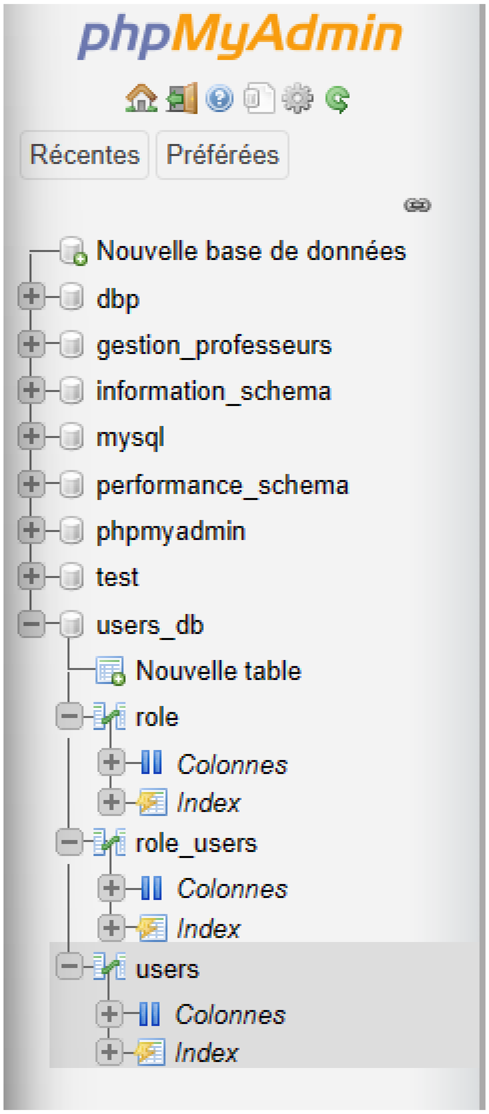


```java
@Override
public void addRoleToUser(String userName, String roleName) {
    User user = findUserByUserName(userName);
    Role role = findRoleByRoleName(roleName);
    if (user.getRoles() != null){
        user.getRoles().add(role);
        role.getUsers().add(user);
    }
}
1 usage
@Override
public User authenticate(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");
}
```

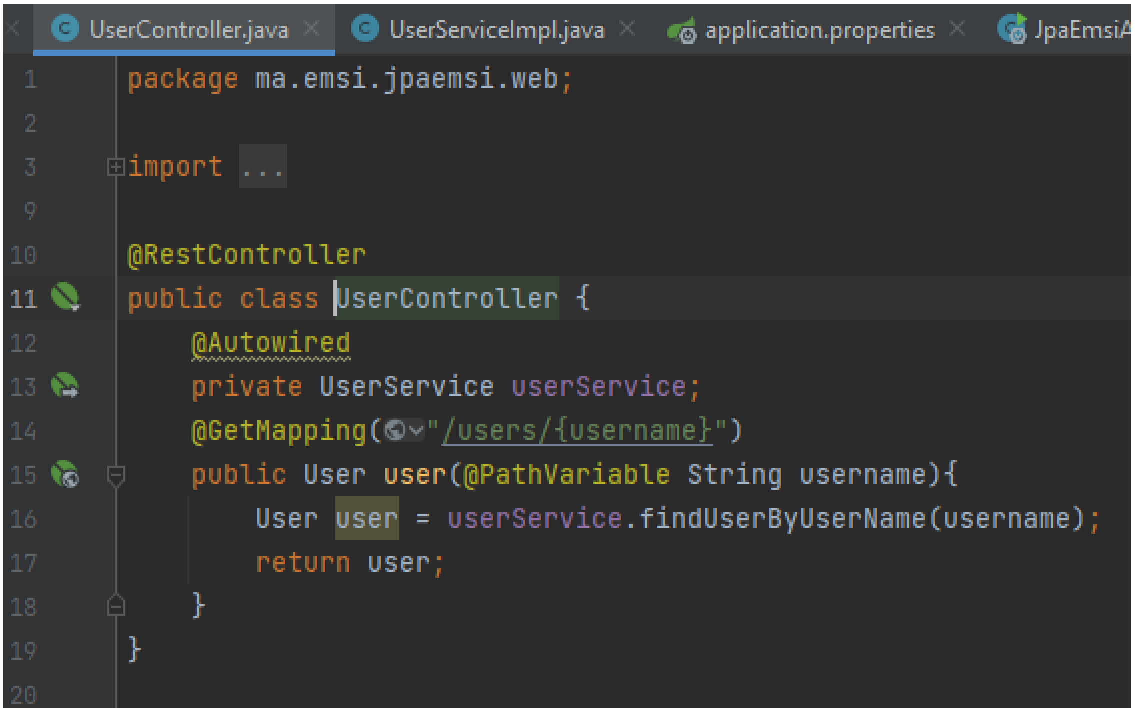
Main



```
ackage ma.emsi.jpaemsi;
mport ...
SpringBootApplication
ublic class JpaEmsiApplication {
   public static void main(String[] args) { SpringApplication.run(JpaEmsiApplication.class,
   @Bean
   CommandLineRunner start(UserService userService) {
       return args -> {
           User u = new User();
           u.setUsername("user1");
           u.setPassword("123456");
           userService.addNewUser(u);

           User u2 = new User();
           u2.setUsername("admin");
           u2.setPassword("123456");
           userService.addNewUser(u2);

           Stream.of( ...values: "STUDENT","USER","ADMIN").forEach(r->{
              Role role1 = new Role();
              role1.setRoleName(r);
              userService.addNewRole(role1);
           });

           userService.addRoleToUser( userName: "user1", roleName: "STUDENT");
           userService.addRoleToUser( userName: "user1", roleName: "USER");
           userService.addRoleToUser( userName: "admin", roleName: "USER");
           userService.addRoleToUser( userName: "admin", roleName: "ADMIN");
           try {
               User user = userService.authenticate( userName: "user1", password: "123456");
               System.out.println(user.getUserId());
               System.out.println(user.getUsername());
               System.out.println("Roles => ");
               user.getRoles().forEach(r->{
                   System.out.println("Role => "+r.toString());
               });
           }catch (Exception e){
               e.printStackTrace();
           }
       };
}
```



User control



```java
package ma.emsi.jpaemsi.web;

import ...

@RestController
public class UserController {
    @Autowired
    private UserService userService;
    @GetMapping("/users/{username}")
    public User user(@PathVariable String username){
        User user = userService.findUserByUserName(username);
        return user;
    }
}
```