Ecole Normale Supérieure de l'Enseignement Technique Mohammedia

Université Hassan II de Casablanca

ENSET

ا لمدرسة العليا لأساتذة التعليم التقني المحمدية

جامعة الحسن الثاني بالدار البيضاء

Département Mathématiques et Informatique

Cycle Ingénieur

« GLSID »

COMPTE-RENDU:
Activité pratique N°3
Hibernate et Spring Data

Réalisé par: Tarmoun Amal

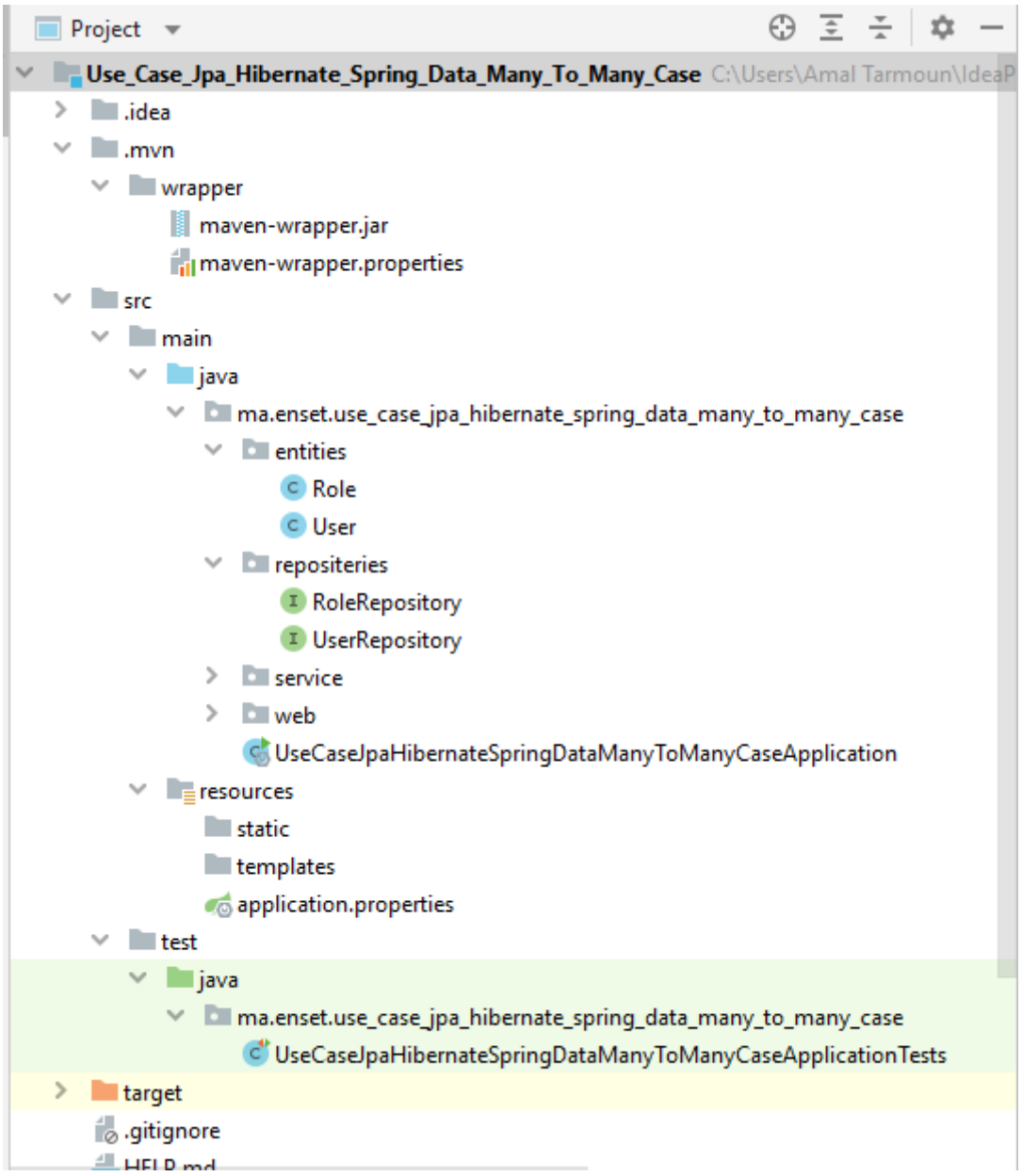
Classe : GLSID2

Encadré par:
Mr. Mohammed EL YOUSSEFI

Année universitaire : 2021 / 2022

Use case jpa hibernate spring data many to many case

1- Cas de Users et Roles :



Le package Entities :

La classe Role

```
6   import lombok.NoArgsConstructor;
7   import lombok.ToString;
8
9   import javax.persistence.*;
10  import java.util.ArrayList;
11  import java.util.List;
12  @Entity
13
14  @Data @NoArgsConstructor @AllArgsConstructor
15  public class Role {
16      @Id  @GeneratedValue(strategy = GenerationType.IDENTITY)
17      private Long id;
18      @Column(name ="DESCRIPTION")
19      private String desc ;
20      @Column(unique = true , length = 20)
21      private String roleName;
22      @ManyToMany(fetch =FetchType.EAGER)
23     // @JoinTable(name =" USERS_ROLES ",)
24      @ToString.Exclude
25      @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
26      private List<User> users = new ArrayList<>() ;
27
28  }
29
```

La classe User :

```java
import lombok.Data;
import lombok.NoArgsConstructor;

import javax.persistence.*;
import java.util.ArrayList;
import java.util.List;
@Entity
@Table(name = "USERS")
@Data @NoArgsConstructor @AllArgsConstructor
public class User {
    @Id
    private String userId ;
    @Column(unique = true, length = 20)

    private String userName;
    private String password;
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    @ManyToMany(mappedBy = "users", fetch= FetchType.EAGER)
    private List<Role> roles = new ArrayList<>();

}
```

Le package Repository

L'interface RoleRepository

```java
package ma.enset.use_case_jpa_hibernate_spring_data_many_to_many_case.repositeries;

import ma.enset.use_case_jpa_hibernate_spring_data_many_to_many_case.entities.Role;
import org.springframework.data.jpa.repository.JpaRepository;

public interface RoleRepository extends JpaRepository<Role,Long> {

    // Role findByRoleName(String roleName);
    Role findByRoleName(String roleName);
}
```

L'interface UserRepository

```java
package ma.enset.use_case_jpa_hibernate_spring_data_many_to_many_case.repositeries;

import ma.enset.use_case_jpa_hibernate_spring_data_many_to_many_case.entities.User;
import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User,String> {
    // c'est spring data :0
    //User findByUserName(String userName);
    User findByUserName(String userName);
}
```

Le package Service

L'interface UserService

```java
package ma.enset.use_case_jpa_hibernate_spring_data_many_to_many_case.service;

import ma.enset.use_case_jpa_hibernate_spring_data_many_to_many_case.entities.Role;
import ma.enset.use_case_jpa_hibernate_spring_data_many_to_many_case.entities.User;

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

La classe UserServiceImpl qui implémente l'interface UserService

```java
package ma.enset.use_case_jpa_hibernate_spring_data_many_to_many_case.service;

import ...

@Service // component de la couche service
@Transactional
public class UserServiceImpl implements IUserService {


    private UserRepository userRepository ;
    private RoleRepository roleRepository ;
    // pour faire l'injection via le constructeur il faut donner a spring un seul constructeur
    // qd il va instancier il va utiliser ce constructeur

    // iinjection via le coonstruucteur.
    public UserServiceImpl(UserRepository userRepository, RoleRepository roleRepository) {
        this.userRepository = userRepository;
        this.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 findRoleByRoleNmae(String roleName) { return roleRepository.findByRoleName(roleName); }

    @Override
    public void addRoleToUser(String username, String roleName) {
        User user  = findUserByUserName(username);
        Role role = findRoleByRoleNmae(roleName);
        // il s'agit d'un update  , car la methode est transactionnel
        user.getRoles().add(role);
        role.getUsers().add(user);

        userRepository.save(user);


    }

}
```

```
61
62      @Override
63      public User authenticate(String userName, String password) {
64          User user = userRepository.findByUserName(userName);
65          if(user==null) throw new RuntimeException("Bad credentials");
66          if(user.getPassword().equals(password)){
67          return user;
68      }
69          throw new RuntimeException("Bad credentials");
70  }}
71
```

Le package web

UserController

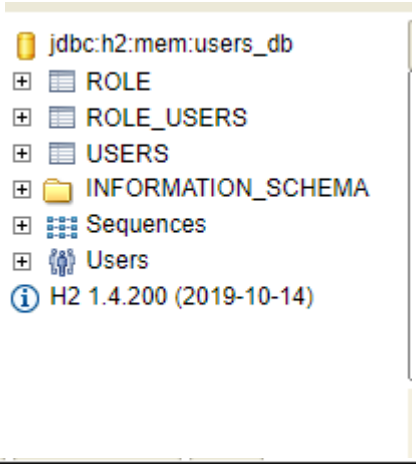
```
1   package ma.enset.use_case_jpa_hibernate_spring_data_many_to_many_case.web;
2
3   import ma.enset.use_case_jpa_hibernate_spring_data_many_to_many_case.entities.User;
4   import ma.enset.use_case_jpa_hibernate_spring_data_many_to_many_case.service.IUserService;
5   import org.springframework.beans.factory.annotation.Autowired;
6   import org.springframework.web.bind.annotation.GetMapping;
7   import org.springframework.web.bind.annotation.PathVariable;
8   import org.springframework.web.bind.annotation.RestController;
9
10  @RestController
11  public class UserController {
12      @Autowired
13      private IUserService userService;
14      @GetMapping("/users/{username}")
15      public User user(@PathVariable  String username){
16          User user =userService.findUserByUserName(username);
17          return user ;
18      }
19  }
20
```

La class JpaManyToManyApplication

```java
package ma.enset.use_case_jpa_hibernate_spring_data_many_to_many_case;

import ...

@SpringBootApplication
public class UseCaseJpaHibernateSpringDataManyToManyCaseApplication {

    public static void main(String[] args) {
        SpringApplication.run(UseCaseJpaHibernateSpringDataManyToManyCaseApplication.class, args);
    }
    @Bean // methode qui s'execute au demmarage
    CommandLineRunner start(IUserService userService) {
        return args -> {
            User u = new User();
            u.setUserName("user1");
            u.setPassword("12345");
            userService.addNewUser(u);

            User u2 = new User();
            u2.setUserName("admin");
            u2.setPassword("12345");
            userService.addNewUser(u2);


            Stream.of("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: "12345");
                System.out.println(user.getUserId());
                System.out.println(user.getUserName());
                System.out.println("Roles=>");
                user.getRoles().forEach(r->{
                    System.out.println("Role =>" +r);
                });

            } catch (Exception e) {
                e.printStackTrace();
            }
        };
    }}
```

Résultat

La base de données H2 :




```
SELECT * FROM ROLE
```

SELECT * FROM ROLE;

ID	DESCRIPTION	ROLE_NAME
1	*null*	STUDENT
2	*null*	USER
3	*null*	ADMIN

(3 rows, 8 ms)

Edit

```
SELECT * FROM USERS
```

SELECT * FROM USERS;

USER_ID	PASSWORD	USER_NAME
6807df91-4855-4681-8f65-99423d48db41	12345	user1
c35b31ec-f259-4123-b66a-cd428c6e5650	12345	admin

(2 rows, 8 ms)

Edit

Run | Run Selected | Auto complete | Clear | SQL statement:

```
SELECT * FROM ROLE_USERS
```

SELECT * FROM ROLE_USERS;

ROLES_ID	USERS_USER_ID
1	6807df91-4855-4681-8f65-99423d48db41
2	6807df91-4855-4681-8f65-99423d48db41
2	c35b31ec-f259-4123-b66a-cd428c6e5650
3	c35b31ec-f259-4123-b66a-cd428c6e5650

(4 rows, 8 ms)

La partie Web :