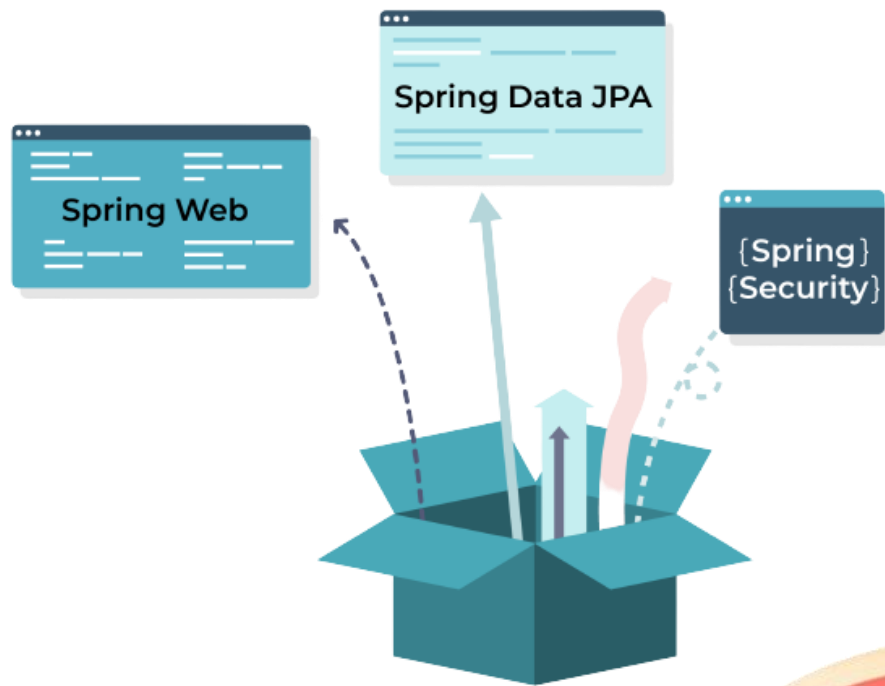


# Compte rendu de :

## SEANCE 05



**Réalisé Par** : AYOUIJIL Soukayna  
**Département** : Mathématiques et Informatique  
**Filière** : II-BDCC  
**Professeur** : M. Mohamed YOUSSEFI

- Gestion des entités :

- Role

```

1  import lombok.NoArgsConstructor;
2  import lombok.ToString;
3
4
5  import javax.persistence.*;
6  import java.util.ArrayList;
7  import java.util.List;
8
9  @Entity
10 @Data @NoArgsConstructor @AllArgsConstructor
11 public class Role {
12     @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
13     private Long id;
14     private String description;
15     @Column(unique = true, length = 20)
16     private String roleName;
17     @ManyToMany(fetch = FetchType.EAGER)
18     @ToString.Exclude
19     @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
20     private List<User> users = new ArrayList<>();
21 }

```

- User

```

1  import lombok.Data;
2  import lombok.NoArgsConstructor;
3
4
5  import javax.persistence.*;
6  import java.util.ArrayList;
7  import java.util.List;
8
9  @Entity
10 @Table(name = "USERS")
11 @Data @NoArgsConstructor @AllArgsConstructor
12 public class User {
13     @Id
14     private String userId;
15     @Column(unique = true, length = 20)
16     private String userName;
17     private String password;
18     @ManyToMany(mappedBy = "users", fetch = FetchType.EAGER)
19     private List<Role> roles = new ArrayList<>();
20 }

```

- Gestion des repositories

- RoleRepository

```

1  package ma.enset.jpaenset.repositories;
2
3  import ma.enset.jpaenset.entities.Role;
4  import org.springframework.data.jpa.repository.JpaRepository;
5
6  public interface RoleRepository extends JpaRepository<Role, Long> {
7      Role findByName(String roleName);
8  }

```

- UserRepository

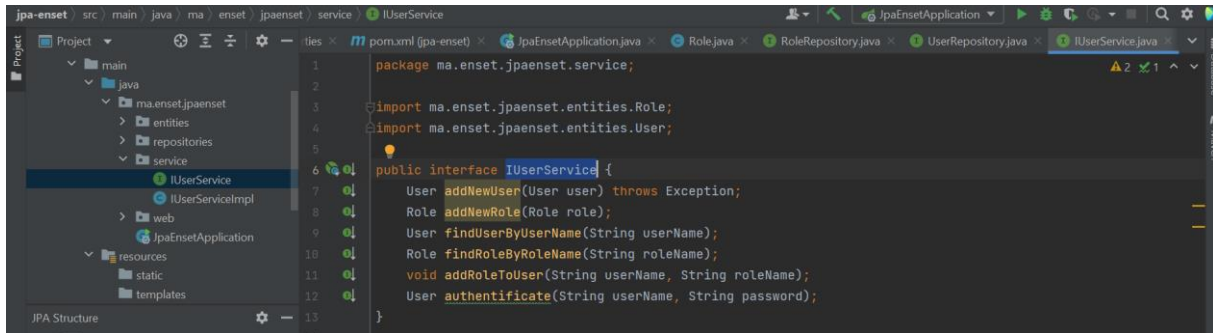
```

1  package ma.enset.jpaenset.repositories;
2
3  import ma.enset.jpaenset.entities.User;
4  import org.springframework.data.jpa.repository.JpaRepository;
5
6  public interface UserRepository extends JpaRepository<User, String> {
7      User findByName(String userName);
8  }

```

- 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.findByName(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);
        }
    }
}
```

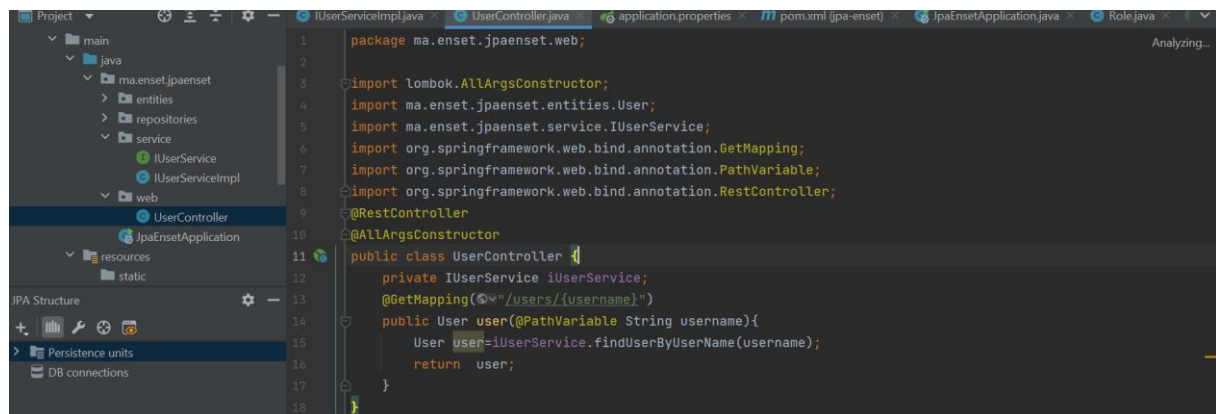
```

        role.getUsers().add(user);
    }
    //userRepository.save(user);
}

@Override
public User 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");
}
}
}

```

- Couche Web



```

1 package ma.enset.jpaenset.web;
2
3 import lombok.AllArgsConstructor;
4 import ma.enset.jpaenset.entities.User;
5 import ma.enset.jpaenset.service.IUserService;
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 @AllArgsConstructor
12 public class UserController {
13     private IUserService iUserService;
14     @GetMapping("/users/{username}")
15     public User user(@PathVariable String username){
16         User user=iUserService.findUserByUserName(username);
17         return user;
18     }
19 }

```

- 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);
        }
    }
}

```

- application.properties

```
application.properties ×
1 #spring.h2.console.enabled=true
2 #spring.datasource.url=jdbc:h2:mem:users_db
3 server.port=8083
4 spring.datasource.url=jdbc:mysql://localhost:3306/users_db?createDatabaseIfNotExist=true
5 spring.datasource.username=root
6 spring.datasource.password=
7 spring.jpa.hibernate.ddl-auto=create
8 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MariaDBDialect
9 spring.jpa.show-sql=true
```

- Resultats :

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0,000 seconde(s).)

`SELECT * FROM `role``

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes : Chercher dans cette table | Trier sur l'index : Aucun(e)

+ Options

		id	descri	role_name
<input type="checkbox"/>	Éditer Copier Supprimer	1	my role is : STUDENT	STUDENT
<input type="checkbox"/>	Éditer Copier Supprimer	2	my role is : USER	USER
<input type="checkbox"/>	Éditer Copier Supprimer	3	my role is : ADMIN	ADMIN

⬆️ ☐ Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes : Chercher dans cette table | Trier sur l'index : Aucun(e)

✓ Affichage des lignes 0 - 3 (total de 4, traitement en 0,0000 seconde(s).)

SELECT \* FROM `role\_users`

Profilage

[Éditer en ligne]

[Éditer]

[Expliquer SQL]

[Créer le code source PHP]

[Actualiser]

☐ Tout afficher

Nombre de lignes : 25

Filtrer les lignes : Chercher dans cette table

Trier sur l'index: Aucun(e)

Options

roles_id	users_user_id
1	be2c6fbc-29fe-4e52-9ed2-ac43e23bdffd
2	be2c6fbc-29fe-4e52-9ed2-ac43e23bdffd
2	8e898aae-6268-4d0b-948f-55b1758686e6
3	8e898aae-6268-4d0b-948f-55b1758686e6

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0,0000 seconde(s).)

SELECT \* FROM `users`

Profilage

[Éditer en ligne]

[Éditer]

[Expliquer SQL]

[Créer le code source PHP]

[Actualiser]

☐ Tout afficher

Nombre de lignes : 25

Filtrer les lignes : Chercher dans cette table

Trier sur l'index: Aucun(e)

+ Options

←T→

user\_id

password

user\_name

<input type="checkbox"/>	✎ Éditer	📋 Copier	🗑 Supprimer	8e898aae-6268-4d0b-948f-55b1758686e6	admin	admin
<input type="checkbox"/>	✎ Éditer	📋 Copier	🗑 Supprimer	be2c6fbc-29fe-4e52-9ed2-ac43e23bdffd	user1	user1