



ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR

Membre de
HONORIS UNITED UNIVERSITIES

ATELIER RÉALISÉ
PAR BENDAGHA
ISMAIL

Cet Atelier a pour but de réaliser
une application web JEE basée sur
spring mvc , spring data jpa , et
thymeleaf , qui permet de gérer les
patients.



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**

Membre de
HONORIS UNITED UNIVERSITIES

Partie 1 (Bases de Spring MVC):

- **Afficher la liste des patients**
 - **Faire la pagination**
- **Chercher les patients par nom**
- **Supprimer un patient si disponible**



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**

Membre de
HONORIS UNITED UNIVERSITIES

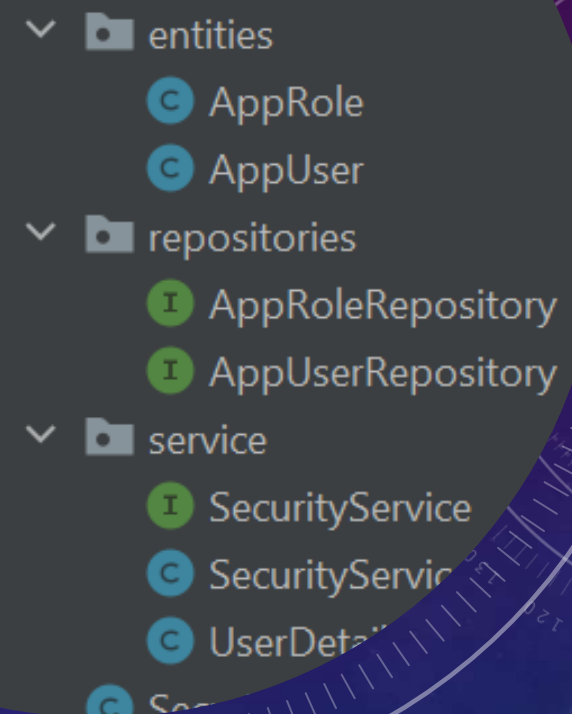
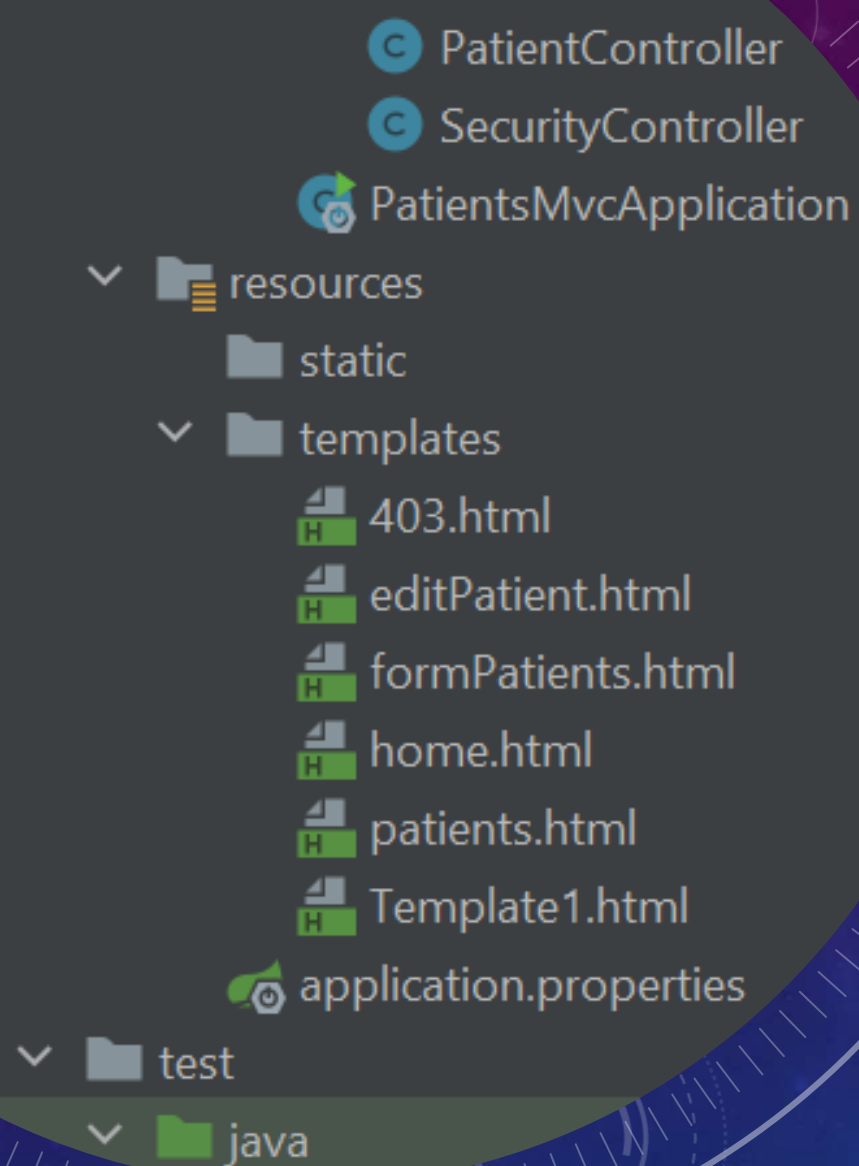
Partie 2 (Thymeleaf Layout et Validation des formulaires):

- **Créer une page template basée sur thymeleaf layout**
 - **Saisir et Ajouter Patients**
 - **Valider le formulaire**
 - **Editer un patient**



Partie 3 & 4 (Spring Security):

- Ajouter la dépendance Maven de Spring Security
- Personnaliser la configuration de Spring Security pour ajouter les restrictions stratégie InMemoryAuthentication
- S'authentifier en tant que USER pour pouvoir enfin de voir la liste des patients
- S'authentifier en tant que ADMIN pour Ajouter, Editer, modifier et supprimer des Patients.
- Permettre l'accès aux ressources statiques sans aucune authentification
- Basculer vers la stratégie UserDetailsService



STRUCTURE DES FICHIERS

- **La page Login**

Please sign in

Sign in

Liste des patients et pagination

Liste des patients

Key word [Chercher](#)

ID	Nom	Homme	Date	malade	adulte	score		
61	Hanae	false	2022-04-26	false	true	196	Delete	Edit
62	Hassan	true	2022-04-26	true	false	126	Delete	Edit
63	Mohammed	true	2022-04-26	false	true	448	Delete	Edit
64	Yasmine	false	2022-04-26	true	true	235	Delete	Edit
65	Hanae	false	2022-04-26	false	true	196	Delete	Edit

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28

Modifier un patient

ID 61

Nom

Hanae

Homme ☐

Date

26/04/2022



Malade ☐

Adulte ☒

score

196

Save

MESSAGE DE CONFIRMATION AVANT SUPPRESION D'UN PATIENT

localhost:8082 indique

êtes-vous sûr?

OK

Annuler

Ajout d'un nouveau patient

Nom

IsmailNouveau

Homme ☒

Date

09/04/2022



Malade ☒

Adulte ☒

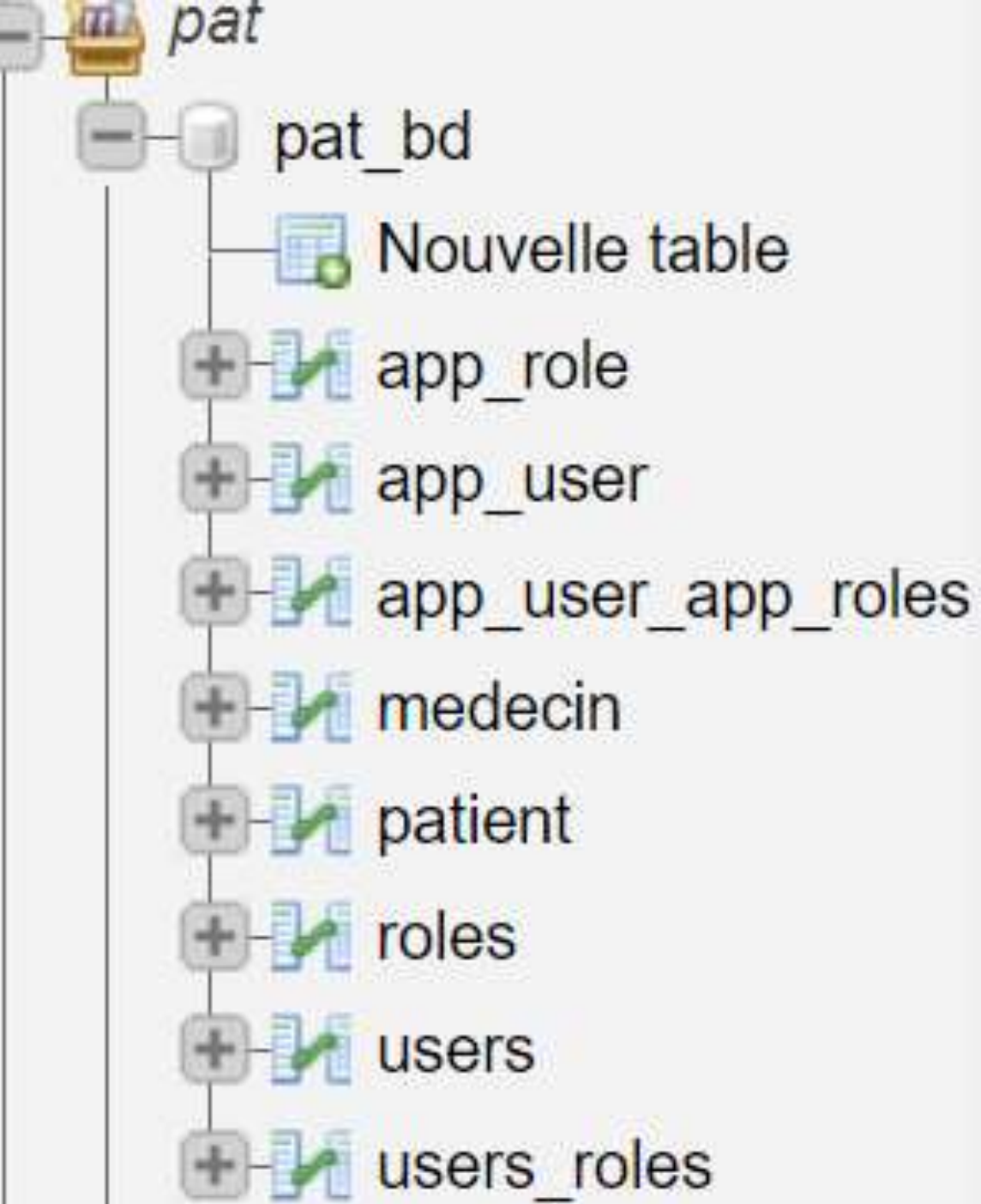
score

256

Save

143	IsmailNouveau	true	2022-04-09	true	true	256	Delete	Edit															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
24	25	26	27	28																			

PATIENT AJOUTE AVEC SUCCES



- **Notre DataBase**

- Les users crees

```
//@Bean
CommandLineRunner saveUsers(SecurityService securityService){
    return args ->{
        securityService.saveNewUser( username: "mohamed", password: "1234", rePassword: "1234");
        securityService.saveNewUser( username: "yasmine", password: "1234", rePassword: "1234");
        securityService.saveNewUser( username: "hassan", password: "1234", rePassword: "1234");

        securityService.saveNewRole( roleName: "USER", description: "");
        securityService.saveNewRole( roleName: "ADMIN", description: "");

        securityService.addRoleToUser( username: "mohamed", roleName: "USER");
        securityService.addRoleToUser( username: "mohamed", roleName: "ADMIN");
        securityService.addRoleToUser( username: "yasmine", roleName: "USER");
        securityService.addRoleToUser( username: "hassan", roleName: "USER");
    };
}
```

- Ceder les roles USER et ADMIN

```
@Override
protected void configure(HttpSecurity http) throws Exception {
    http.formLogin();
    http.authorizeRequests().antMatchers(...antPatterns: "/").permitAll();
    http.authorizeRequests().antMatchers(...antPatterns: "/admin/**").hasAuthority("ADMIN");
    http.authorizeRequests().antMatchers(...antPatterns: "/user/**").hasAuthority("USER");
    http.authorizeRequests().antMatchers(...antPatterns: "/webjars/**").permitAll();
    http.authorizeRequests().anyRequest().authenticated();
    http.exceptionHandling().accessDeniedPage( accessDeniedUrl: "/403");
}
```


Security Service

```
package ma.emsi.patientsmvc.sec.service;
```

```
import ma.emsi.patientsmvc.sec.entities.AppRole;
```

```
import ma.emsi.patientsmvc.sec.entities.AppUser;
```

5 usages 1 implementation

```
public interface SecurityService {
```

3 usages 1 implementation

```
AppUser saveNewUser(String username, String password, String rePassword);
```

2 usages 1 implementation

```
AppRole saveNewRole(String roleName, String description);
```

4 usages 1 implementation

```
void addRoleToUser(String username, String roleName);
```

1 implementation

```
void removeRoleFromUser(String username, String roleName);
```

1 usage 1 implementation

```
AppUser loadUserByUserName(String username);
```

```
}
```


Ajouter user et role

```
@Override
public AppUser saveNewUser(String username, String password, String rePassword) {
    if (!password.equals(rePassword)) throw new RuntimeException("Passwords not matching");
    String hashedPWD=passwordEncoder.encode(password);
    AppUser appUser=new AppUser();
    appUser.setUserId(UUID.randomUUID().toString());
    appUser.setUsername(username);
    appUser.setPassword(hashedPWD);
    appUser.setActive(true);
    AppUser savedAppUser=appUserRepository.save(appUser);
    return savedAppUser;
}
```

2 usages

```
@Override
public AppRole saveNewRole(String roleName, String description) {
    AppRole appRole=appRoleRepository.findByRoleName(roleName);
    if (appRole!=null) throw new RuntimeException("Role" +roleName+"Already exists");
    appRole=new AppRole();
    appRole.setRoleName(roleName);
    appRole.setDescription(description);
    AppRole savedAppRole=appRoleRepository.save(appRole);
    return savedAppRole;
}
```

Modifier le role a un certain user

4 usages

@Transactional

@Override

```
public void addRoleToUser(String username, String roleName) {  
    AppUser appUser=appUserRepository.findByUsername(username);  
    if (appUser==null) throw new RuntimeException("User not found");  
    AppRole appRole=appRoleRepository.findByRoleName(roleName);  
    if (appRole==null) throw new RuntimeException("Role not found");  
    appUser.getAppRoles().add(appRole);  
}
```

@Override

```
public void removeRoleFromUser(String username, String roleName) {  
    AppUser appUser=appUserRepository.findByUsername(username);  
    if (appUser==null) throw new RuntimeException("User not found");  
    AppRole appRole=appRoleRepository.findByRoleName(roleName);  
    if (appRole==null) throw new RuntimeException("Role not found");  
    appUser.getAppRoles().remove(appRole);  
}
```

Github Repository

<https://github.com/ismailbendagha/Atelier-3-4>