

# Activité pratique N°3

Mohamed Farhi 4eme MIAE G1M

## Introduction :

Développer une application web basée sur Spring MVC, Thymeleaf et Spring Data qui permet de gérer des patients.

L'application doit permettre de:

- chercher des patients avec pagination
- supprimer un Patient
- Ajouter un patient avec validation du formulaire
- Editer et mettre à jour un patient
- créer une page template basée sur Thymeleaf Layout

Sécuriser l'accès à l'application qui permet de gérer les Patients en utilisant Spring security:

**1-Ajouter la dépendance maven de Spring security**

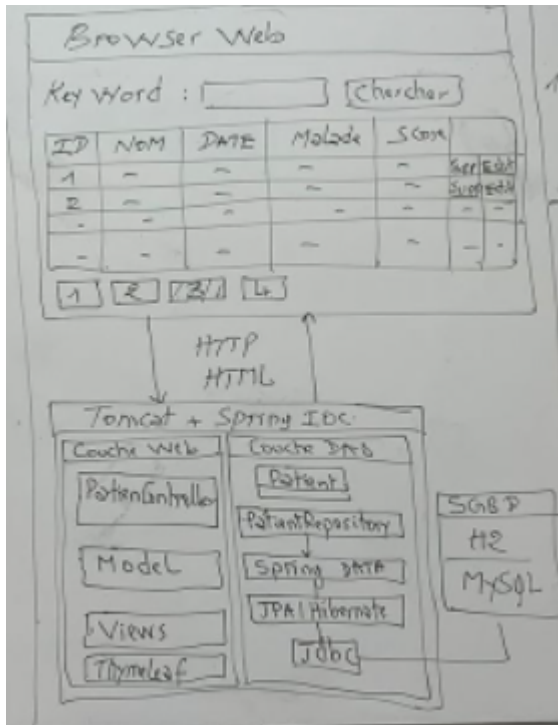
**2- Personnaliser la configuration de Spring security pour ajouter les restrictions suivantes avec la stratégie InMemoryAuthentication**

- Authentification avec le rôle USER pour pouvoir consulter les patients
- Authentification avec le rôle ADMIN pour pouvoir Ajouter, Éditer et Supprimer des Patients

**3-Basculer de la stratégie InMemoryAuthentication vers JDBCAuthentication**

- Stratégie : UserDetailsService
- Améliorations

# Architecture et conception :



## Code source:

GitHub:

<https://github.com/josuke08/Mohamed-Farhi-AP3.git>

## Screens

### Partie 1:

afficher les patients avec pagination

Liste des patients					
Key word		<input type="text"/>			
		<input type="button" value="Chercher"/>			
ID	Nom	Date	Malade	Score	
1	manal	2022-04-11	false	12	<input type="button" value="Delete"/>
2	Mohamed	2022-04-11	true	12	<input type="button" value="Delete"/>
27	Yazid	2022-04-11	true	12	<input type="button" value="Delete"/>
29	manal	2022-04-11	false	12	<input type="button" value="Delete"/>
31	Yazid	2022-04-11	true	12	<input type="button" value="Delete"/>
<div><input type="button" value="0"/> <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/> <input type="button" value="6"/> <input type="button" value="7"/> <input type="button" value="8"/> <input type="button" value="9"/> <input type="button" value="10"/> <input type="button" value="11"/> <input type="button" value="12"/> <input type="button" value="13"/> <input type="button" value="14"/> <input type="button" value="15"/> <input type="button" value="16"/> <input type="button" value="17"/> <input type="button" value="18"/> <input type="button" value="19"/></div>					

recherche des patients:

Liste des patients					
Key word		<input type="text" value="moha"/> <input type="button" value="Chercher"/>			
ID	Nom	Date	Malade	Score	
2	Mohamed	2022-04-11	true	12	<input type="button" value="Delete"/>
46	Mohamed	2022-04-11	true	12	<input type="button" value="Delete"/>
50	Mohamed	2022-04-11	true	12	<input type="button" value="Delete"/>
58	Mohamed	2022-04-11	true	12	<input type="button" value="Delete"/>
62	Mohamed	2022-04-11	true	12	<input type="button" value="Delete"/>
<input type="button" value="0"/> <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/>					

supprimer un patient(id = 2):

Liste des patients					
Key word		<input type="text" value="moha"/> <input type="button" value="Chercher"/>			
ID	Nom	Date	Malade	Score	
46	Mohamed	2022-04-11	true	12	<input type="button" value="Delete"/>
50	Mohamed	2022-04-11	true	12	<input type="button" value="Delete"/>
58	Mohamed	2022-04-11	true	12	<input type="button" value="Delete"/>
62	Mohamed	2022-04-11	true	12	<input type="button" value="Delete"/>
66	Mohamed	2022-04-11	true	12	<input type="button" value="Delete"/>
<input type="button" value="0"/> <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/>					

Partie 2:

Template basée sur thymeleaf layout

Logo
Home
Link
Disabled
Patients
[Username]

Liste des patients

Key word
Chercher

ID	Nom	Date	Malade	Score		
1	manal	2022-04-11	false	12	Delete	Edit
27	Yazid	2022-04-11	true	12	Delete	Edit
29	manal	2022-04-11	false	12	Delete	Edit
31	Yazid	2022-04-11	true	12	Delete	Edit
33	manal	2022-04-11	false	12	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
29

formulaire pour ajouter un patient

Patients
[Username]

Nouveaux
Chercher

Logo
Home
Link
Disabled
Patients
[Username]

Nom:
Date Naissance:
mm/dd/yyyy
Malade:
Score:
0
Save

modifier un patient

---

ID: 178

Nom:

Suttony

Date Naissance:

mm/dd/yyyy



Malade: ☐

Score:

777

Save

## 1-Ajouter la dépendance maven de Spring security

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

## 2- Personnaliser la configuration de Spring security pour ajouter les restrictions suivantes avec la stratégie InMemoryAuthentication

```
@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
    PasswordEncoder passwordEncoder = passwordEncoder();
    String encodedPWD = passwordEncoder.encode( rawPassword: "1234");
    System.out.println(encodedPWD);
    auth.inMemoryAuthentication()
        .withUser( username: "user1").password(encodedPWD).roles("USER")
        .and()
        .withUser( username: "user2").password(passwordEncoder.encode( rawPassword: "1111")).roles("USER")
        .and()
        .withUser( username: "admin").password(passwordEncoder.encode( rawPassword: "2345")).roles("USER", "ADMIN");
}

@Bean
PasswordEncoder passwordEncoder(){
    return new BCryptPasswordEncoder();
}
```

-Authentication avec le rôle USER pour pouvoir consulter les patients

Please sign in

Sign in

Liste des patients

Key word

Chercher

ID	Nom	Date	Malade	Score
62	Mohamed	2022-04-11	true	155
178	Suttony		false	777
179	manal	2022-04-12	false	120
180	Mohamed	2022-04-12	true	120
181	Yazid	2022-04-12	true	120

0

1

2

3

-Authentification avec le rôle ADMIN pour pouvoir Ajouter, Éditer et Supprimer des Patients

# Please sign in

You have been signed out

admin

....|

Sign in

Liste des patients

Key word  [Chercher](#)

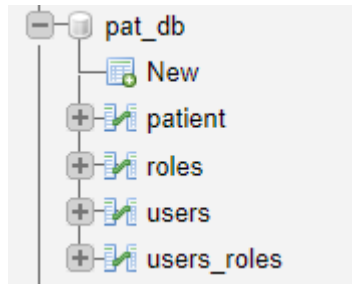
ID	Nom	Date	Malade	Score		
62	Mohamed	2022-04-11	true	155	<a href="#">Delete</a>	<a href="#">Edit</a>
179	manal	2022-04-12	false	120	<a href="#">Delete</a>	<a href="#">Edit</a>
180	Mohamed	2022-04-12	true	120	<a href="#">Delete</a>	<a href="#">Edit</a>
181	Yazid	2022-04-12	true	120	<a href="#">Delete</a>	<a href="#">Edit</a>
182	salomon	2022-04-12	false	120	<a href="#">Delete</a>	<a href="#">Edit</a>

0 1 2 3

### 3-Basculer de la stratégie InMemoryAuthentication vers JDBCAuthentication

```
32 auth.jdbcAuthentication()  
33     .dataSource(dataSource)  
34     .usersByUsernameQuery("select username as principal, password as credentials, active from users where username=?")  
35     .authoritiesByUsernameQuery("Select username as principal, role as role from users_roles where username=?")  
36     .rolePrefix("ROLE_")  
37     .passwordEncoder(passwordEncoder);
```

phpMyAdmin:



users:

+ Options				username	password	active
<input type="checkbox"/>				admin	\$2a\$10\$JcNRcylveZzz6FCNcY3pyOGR769kk74vUGYZrk8BM6O...	1
<input type="checkbox"/>				user1	\$2a\$10\$JcNRcylveZzz6FCNcY3pyOGR769kk74vUGYZrk8BM6O...	1
<input type="checkbox"/>				user2	\$2a\$10\$JcNRcylveZzz6FCNcY3pyOGR769kk74vUGYZrk8BM6O...	1

roles:

+ Options				role
<input type="checkbox"/>				ADMIN
<input type="checkbox"/>				USER

users\_roles:

+ Options				username	role
<input type="checkbox"/>				admin	ADMIN
<input type="checkbox"/>				admin	USER
<input type="checkbox"/>				user1	ADMIN
<input type="checkbox"/>				user1	USER
<input type="checkbox"/>				user2	USER

Stratégie : UserDetailsService



```
auth.userDetailsService(userDetailsService);
```

L'interface SecurityService et les entités:

- AppRole
- AppUser

```
public interface SecurityService {  
    AppUser saveNewUser(String username, String password, String rePassword);  
    AppRole saveNewRole(String roleName, String description);  
    void addRoleToUser(String username, String roleName);  
    void removeRoleToUser(String username, String roleName);  
    AppUser loadUserByUsername(String username);  
}
```

Population des 2 entités

```
CommandLineRunner saveUsers(SecurityService securityService){  
    return args -> {  
        securityService.saveNewUser( username: "mohamed", password: "1234", rePassword: "1234");  
        securityService.saveNewUser( username: "momo", password: "1234", rePassword: "1234");  
        securityService.saveNewUser( username: "sut", 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: "momo", roleName: "USER");  
        securityService.addRoleToUser( username: "sut", roleName: "USER");  
    };  
}
```

l'implémentation de UserDetailsService

```

package com.example.atelier3.Security.service;

import ...

@Service
public class UserDetailsServiceImpl implements UserDetailsService {

    @Autowired
    private SecurityService securityService;

    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        AppUser appUser = securityService.loadUserByUsername(username);
        /*Collection<GrantedAuthority> authorities = new ArrayList<>();
        appUser.getAppRoles().forEach(role -> {
            SimpleGrantedAuthority authority = new SimpleGrantedAuthority(role.getRoleName());
            authorities.add(authority);
        });*/
        Collection<GrantedAuthority> authorities = appUser.getAppRoles().stream()
            .map(role -> new SimpleGrantedAuthority(role.getRoleName()))
            .collect(Collectors.toList());
        User user = new User(appUser.getUsername(), appUser.getPassword(), authorities);
        return user;
    }
}

```

Améliorations:

Recherche Multi Critère:

j'ai ajouté la date de naissance et s'il est malade ou non

Liste des patients

Key word

Malade ☒

Date

ID	Nom	CIN	Date	Malade	Score		
219	Yazid	waadfmqdlsgsg54	2022-04-26	true	120	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>

Ajouter d'autres attributs à la classe Patient:

j'ai ajouté l'attribut CIN

```

@Entity
@Data @AllArgsConstructor @NoArgsConstructor
public class Patient {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @NotEmpty
    @Size(min=4, max=40)
    private String nom;
    @Column(unique = true)
    private String cin;
    @Temporal(TemporalType.DATE)
    @DateTimeFormat(pattern = "yyy-MM-dd")
    private Date dateNaissance;
    private boolean malade;
    @DecimalMin("100")
    private int score;
}

```

## corriger quelques imperfections

j'ai modifié la fonction d'attribution de rôles afin d'éviter l'affectation un rôle plusieurs fois au même utilisateur

```

@Override
public void addRoleToUser(String username, String roleName) {
    AppUser appUser = appUserRepository.findByUsername(username);
    if(appUser == null) throw new RuntimeException("User " + username + " doesn't exists");
    AppRole appRole = appRoleRepository.findByRoleName(roleName);
    if(appRole == null) throw new RuntimeException("Role " + roleName + " doesn't exists");

    if(!appUser.containsRole(appRole))
        appUser.getAppRoles().add(appRole);
}

```

voila le code source de la méthode containsRole(AppRole):

```

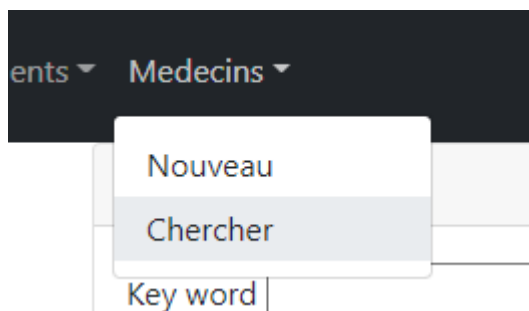
public boolean containsRole(AppRole role) {
    for(AppRole r: this.appRoles){
        if (r.getRoleName().equals(role.getRoleName()))
            return true;
    }
    return false;
}

```

## Ajouter la gestion de l'entité Medecin:

j'ai ajouté une entité Medecin et les fonctionnalités d'ajout, modification, suppression et affichage de la liste des médecins:

```
@Entity
@Data @AllArgsConstructor @NoArgsConstructor
public class Medecin {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @NotEmpty
    @Size(min=4, max=40)
    private String nom;
    @Column(unique = true)
    private String cin;
    @NotEmpty
    private String specialite;
}
```



ID	Nom	CIN	Specialité		
1	Hippocrates	kkl	cardiologie	Delete	Edit
4	Sutton	fl	cardiologie	Delete	Edit
6	Hippocrate	kakl	cardiologie	Delete	Edit
7	TrishaPaytasa	kadl	cardiologie	Delete	Edit
9	Sutton	fal	cardiologie	Delete	Edit
13	Frank	ksasl	cardiologie	Delete	Edit
14	Sutton	fasl	cardiologie	Delete	Edit
15	Hippocrate	ksakdl	cardiologie	Delete	Edit
16	TrishPaytas	ksadld	cardiologie	Delete	Edit
17	Frank	ksasld	cardiologie	Delete	Edit

0

1

Nom

CIN

Specialité

Save

Ajouter l'entité RendezVous

```
@Entity
@Data @AllArgsConstructor @NoArgsConstructor
public class RendezVous {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Temporal(TemporalType.DATE)
    @DateTimeFormat(pattern = "yyy-MM-dd")
    private Date date;
    private String description;
    @OneToOne(fetch = FetchType.EAGER)
    private Medecin medecin;
    @OneToOne(fetch = FetchType.EAGER)
    private Patient patient;
}
```