

## DÉPARTEMENT MATHÉMATIQUES ET INFORMATIQUE

**Filière :**  
**« Ingénierie Informatique : Big Data et Cloud Computing »**  
**II-BDCC**

### **TP 3 : Spring MVC Thymeleaf**

**Réalisé Par:** Laila Sad Elbouyoud

**Classe:** ii-BDCC 2

# Interface Graphique

## Page Login:

Please sign in

Sign in

## Page d'accueil de l'utilisateur qui possède un role Admin:

Accueil Link Link Disabled Patients ▾

laila ▾

Liste des Patients

keyword

Chercher

ID	Nom	Date	Malade	Score		
1	Hassan	2022-04-01	false	12	Supprimer	Modifier
2	Mohammed	2022-04-01	true	122	Supprimer	Modifier
3	Laila	2022-04-01	true	342	Supprimer	Modifier
4	Asmaa	2022-04-01	false	123	Supprimer	Modifier
5	Hassan	2022-04-01	false	12	Supprimer	Modifier

0

1

2

3

4

5

6

7

8

9

## Page d'accueil pour les utilisateurs qui ne sont pas des Admins:

[Accueil](#) [Link](#) [Link](#) [Disabled](#) [Patients](#) [▼](#)ilham ▼

Liste des Patients

keyword  [Chercher](#)

ID	Nom	Date	Malade	Score
1	Hassan	2022-04-01	false	12
2	Mohammed	2022-04-01	true	122
3	Laila	2022-04-01	true	342
4	Asmaa	2022-04-01	false	123
5	Hassan	2022-04-01	false	12

[0](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#)

L'admin peut Rechercher, Ajouter, modifier et supprimer les patients. Pendant que les utilisateurs normaux peuvent seulement chercher et naviguer dans les pages des patients.

## Page de déconnexion:

Are you sure you want to log out?

[Log Out](#)

## Page de Modification:

Accueil Link Link Disabled Patients ▼

laila ▼

ID : 3

Nom

Laila

Date Naissance

04/01/2022

Malade ☒

Score

342

Soumettre

## Suppression:

- Message d'alerte:

Disabled Patients ▼

localhost:8082 says

Etes vous surs?

OK

Cancel

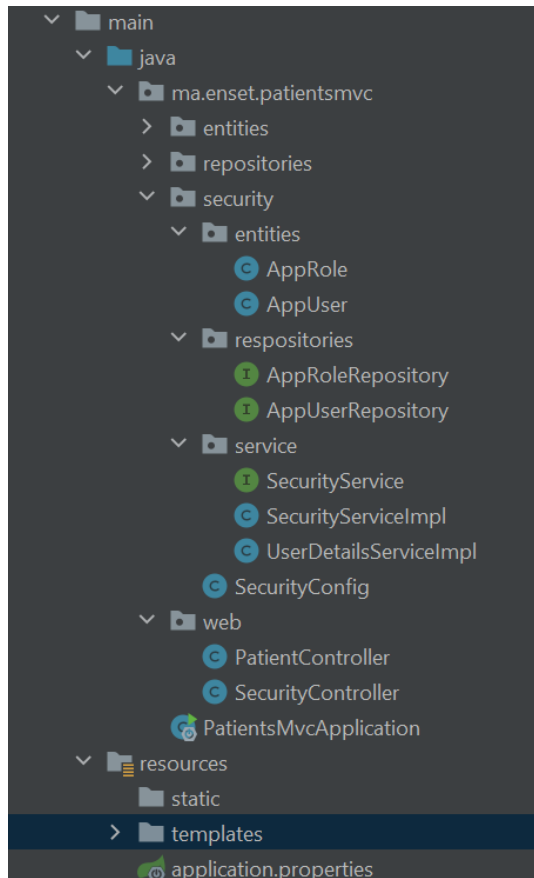
Patients

Chercher

Nom	Date	Malade	Score		
Hassan	2022-04-01	false	12	Supprimer	Modifier
Mohammed	2022-04-01	true	122	Supprimer	Modifier
Laila	2022-04-01	true	342	Supprimer	Modifier
Asmaa	2022-04-01	false	122	Supprimer	Modifier

# Partie Code

## La structure du projet:



## Les entités:

### Patient:

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

## Respositories:

### PatientRespository:

```
public interface PatientRepository extends JpaRepository<Patient, Long> {  
    Page<Patient> findByNomContains(String kw, Pageable pageable);  
}
```

## Security:

### entitiés:

#### AppRole:

```
@Entity  
@Data  
@AllArgsConstructor @NoArgsConstructor  
public class AppRole {  
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long roleId;  
    @Column(unique = true)  
    private String roleName;  
    private String description;  
}
```

#### AppUser:

```
@Entity  
@Data  
@AllArgsConstructor @NoArgsConstructor  
public class AppUser {  
    @Id  
    private String userId;  
    @Column(unique = true)  
    private String username;  
    private String password;  
    private boolean active;  
    @ManyToMany(fetch = FetchType.EAGER)  
    private List<AppRole> appRoles = new ArrayList<>();  
}
```

### Respositories:

#### AppRoleRepository:

```
public interface AppRoleRepository extends JpaRepository<AppRole, Long> {  
    AppRole findByRoleName(String roleName);  
}
```

#### AppUserRepository:

```
public interface AppUserRepository extends JpaRepository<AppUser, String> {  
    AppUser findByUsername(String username);  
}
```

### Service:

#### Interface SecurityService:

```
public interface SecurityService {  
    AppUser saveUser(String username, String password, String rePassword);  
    AppRole addNewRole(String roleName, String description);  
    void addRoleToUser(String username, String roleName);  
    void removeRoleFromUser(String username, String roleName);  
    AppUser loadUserByUsername(String username);  
}
```

#### Implémentation de SecurityService:

```
@Service  
@Slf4j  
@AllArgsConstructor  
@Transactional  
public class SecurityServiceImpl implements SecurityService {  
    private AppUserRepository appUserRepository;  
    private AppRoleRepository appRoleRepository;  
    private PasswordEncoder passwordEncoder;  
  
    @Override  
    public AppUser saveUser(String username, String password, String rePassword) {  
        if(!password.equals(rePassword)) throw new RuntimeException("Les mots de passe ne correspondent");  
        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;  
    }  
}
```

## Implémentation de UserDetailsService de Spring:

```
@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> authorities1 =
            appUser.getAppRoles().stream()
                .map(role -> new SimpleGrantedAuthority(role.getRoleName()))
                .collect(Collectors.toList());
        User user = new User(appUser.getUsername(), appUser.getPassword(), authorities1);
        return user;
    }
}
```

## SecurityConfig:

```
auth.userDetailsService(userDetailsService);

}

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



## Web:

### PatientController:

```
@Controller
@AllArgsConstructor
public class PatientController {
    private PatientRepository patientRepository;

    @GetMapping(path = "/user/index")
    public String patients(Model model,
        @RequestParam(name="page", defaultValue = "0") int page,
        @RequestParam(name="size", defaultValue = "5") int size,
        @RequestParam(name="keyword", defaultValue = "") String keyword){
        Page<Patient> pagePatients = patientRepository.findByNomContains(keyword, PageRequest.of(page, size));
        model.addAttribute("listPatients", pagePatients.getContent());
        model.addAttribute("pages", new int[pagePatients.getTotalPages()]);
        model.addAttribute("currentPage", page);
        model.addAttribute("keyword", keyword);
        return "patient";
    }
}
```

### SecurityController:

```
@Controller
public class SecurityController {
    @GetMapping("/403")
    public String notAuthorized(){
        return "403";
    }
}
```

## PatientsMVCApplication:

```
//@Bean
CommandLineRunner saveUser(SecurityService securityService){
    return args -> {
        securityService.saveUser( username: "laila", password: "1234", rePassword: "1234");
        securityService.saveUser( username: "ilham", password: "1234", rePassword: "1234");
        securityService.saveUser( username: "hafsa", password: "1234", rePassword: "1234");

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

        securityService.addRoleToUser( username: "laila", roleName: "USER");
        securityService.addRoleToUser( username: "laila", roleName: "ADMIN");
        securityService.addRoleToUser( username: "ilham", roleName: "USER");
        securityService.addRoleToUser( username: "hafsa", roleName: "USER");
    };
}
```

## Templates:

### 403.html:

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
      xmlns:sec="http://www.thymeleaf.org/extras/spring-security"
      layout:decorate="template1">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>

<div layout:fragment="content1">
  <div class="container">
    <h2 class="text-danger">Vous n'êtes pas autorisés à accéder à cette ressource</h2>
  </div>
</body>
</html>
```

### editPatient.html:

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
      layout:decorate="template1">
<head>
  <meta charset="UTF-8">
  <title>Modifier Patient</title>
  <link rel="stylesheet" href="/webjars/bootstrap/5.1.3/css/bootstrap.min.css"/>
</head>
<body>
<div layout:fragment="content1">
  <div class="col-md-6 offset-3">
    <form method="post" th:action="@{/admin/save(page=${page}, keyword=${keyword})}">
      <div>
        <label>ID : </label>
        <label th:text="${patient.id}"></label>
        <input class="form-control" type="hidden" name="id" th:value="${patient.id}">
      </div>
      <div>
        <label for="nom">Nom</label>
        <input id="nom" class="form-control" type="text" name="nom" th:value="${patient.nom}">
        <span class="text-danger" th:errors="${patient.nom}"></span>
      </div>
    </form>
  </div>
</div>
```

### formPatients.html:

```
<body>
<div layout:fragment="content1">
  <div class="col-md-6 offset-3">
    <form method="post" th:action="@{/admin/save}">
      <div>
        <label for="nom">Nom</label>
        <input id="nom" class="form-control" type="text" name="nom" th:value="${patient.nom}">
        <span class="text-danger" th:errors="${patient.nom}"></span>
      </div>
      <div>
        <label>Date Naissance</label>
        <input class="form-control" type="date" name="dateNaissance" th:value="${patient.dateNaissance}">
        <span class="text-danger" th:errors="${patient.dateNaissance}"></span>
      </div>
      <div>
        <label>Malade</label>
        <input type="checkbox" name="malade" th:checked="${patient.malade}">
        <span class="text-danger" th:errors="${patient.malade}"></span>
      </div>
      <div>
        <label>Score</label>
        <input class="form-control" type="text" name="score" th:value="${patient.score}">
      </div>
    </form>
  </div>
</div>
```

### home.html:

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org"
  xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
  xmlns:sec="http://www.thymeleaf.org/extras/spring-security"
  layout:decorate="template1">
<head>
  <meta charset="UTF-8">
  <title>Index</title>
</head>
<body>
  <div layout:fragment="content1">
  </div>
</body>
</html>
```

### patient.html:

```
<table class="table">
  <thead>
    <tr>
      <th>ID</th>
      <th>Nom</th>
      <th>Date</th>
      <th>Malade</th>
      <th>Score</th>
    </tr>
  </thead>
  <tbody>
    <tr th:each="p:${listPatients}">
      <td th:text="${p.id}"></td>
      <td th:text="${p.nom}"></td>
      <td th:text="${p.dateNaissance}"></td>
      <td th:text="${p.malade}"></td>
      <td th:text="${p.score}"></td>
      <td sec:authorize="hasAuthority('ADMIN')">
        <a onclick="return confirm('Etes vous surs?')" class="btn btn-danger" th:href="#">
          Supprimer
        </a>
      </td>
    </tr>
  </tbody>
</table>
```

### template1.html:

```
<script src="/webjars/bootstrap/5.1.3/js/bootstrap.bundle.js"></script>
</head>
<body>
  <nav class="navbar navbar-expand-sm bg-light navbar-light">
    <div class="container-fluid">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" th:href="@{/user/index}">Accueil</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled" href="#">Disabled</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown">
```

### Application properties:

```
spring.datasource.url=jdbc:mysql://localhost:3306/db_patients?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=
server.port=8082
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto= update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MariaDBDialect
spring.thymeleaf.cache=false
#spring.main.allow-circular-references=true
```