

DEPARTEMENT MATHÉMATIQUES ET INFORMATIQUE

# Compte rendu

- Travaux Pratique : 3-

-

« Architecture JEE »

SPRING  
MVC THYMELEAF

Réalisé par :

Ikram Berradi

Encadré par :

Mohamed YOUSSEFI

GLSID II-2022/2021

## ❖ Exercice : application Web JEE basée sur Spring MVC, Thymeleaf et Spring Data JPA

Création d'une application Web JEE basée sur Spring MVC, Thymeleaf et Spring Data JPA qui permet de gérer les patients. L'application doit permettre les fonctionnalités suivantes :

- ✓ Afficher les patients
- ✓ Faire la pagination
- ✓ Chercher les patients
- ✓ Supprimer un patient
- ✓ Faire des améliorations supplémentaires

### ➤ Application.properties

```
application.properties
1 #spring.datasource.url=jdbc:h2:mem:patients_db
2 #spring.h2.console.enabled=true
3 spring.datasource.url=jdbc:mysql://localhost:3306/patients_db?createDatabaseIfNotExist=true
4 spring.datasource.username=root
5 spring.datasource.password=
6 spring.jpa.hibernate.ddl-auto=update
7 spring.jooq.sql-dialect=org.hibernate.dialect.MariaDBDialect
8 server.port=8082
9 spring.jpa.show-sql=true
10 spring.thymeleaf.cache=false
11 #spring.main.allow-circular-references=true
```

### ➤ Application Spring Boot

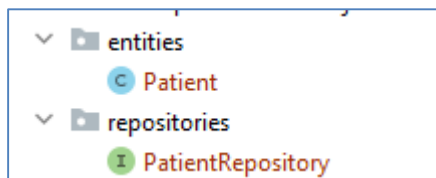
```
2 import ...
14
15 @SpringBootApplication
16 public class PatientsMvcApplication {
17
18     public static void main(String[] args) { SpringApplication.run(PatientsMvcApplication.class, args); }
19
20     @Bean
21     PasswordEncoder passwordEncoder() { return new BCryptPasswordEncoder(); }
22     // @Bean
23     CommandLineRunner start(PatientRepository patientRepository){
24         return args->{
25             Stream.of("Ahlam", "Salma", "Amjad", "Mohammed").forEach(nom->{
26                 patientRepository.save(new Patient(id: null, nom, new Date(), Math.random()<0.5?false:true, score: 129));
27             });
28             patientRepository.findAll().forEach(p->{
29                 System.out.println(p.getNom());
30             });
31         };
32     }
33 }
34 // @Bean
35 CommandLineRunner saveUsers(ISecurityService iSecurityService){
36     return args -> {
37         iSecurityService.saveNewUser(username: "ikram", password: "1234", rePassword: "1234");
38         iSecurityService.saveNewUser(username: "Akram", password: "1234", rePassword: "1234");
39         iSecurityService.saveNewUser(username: "Hassan", password: "1234", rePassword: "1234");
40     }
41 }
```

```

36 // @Bean
37 CommandLineRunner saveUsers(ISecurityService iSecurityService){
38     return args -> {
39         iSecurityService.saveNewUser( username: "ikram", password: "1234", rePassword: "1234");
40         iSecurityService.saveNewUser( username: "Akram", password: "1234", rePassword: "1234");
41         iSecurityService.saveNewUser( username: "Hassan", password: "1234", rePassword: "1234");
42
43         iSecurityService.saveNewRole( roleName: "ADMIN", description: "");
44         iSecurityService.saveNewRole( roleName: "USER", description: "");
45
46         iSecurityService.addRoleToUser( username: "ikram", roleName: "ADMIN");
47         iSecurityService.addRoleToUser( username: "ikram", roleName: "USER");
48         iSecurityService.addRoleToUser( username: "Akram", roleName: "USER");
49         iSecurityService.addRoleToUser( username: "Hassan", roleName: "USER");
50     };
51 }
52
53 }

```

## ➤ Couche DAO



### ➤ Entité JPA : Patient

```

3 import ...
13
14 @Entity
15 @Data @NoArgsConstructor @AllArgsConstructor
16 public class Patient
17 {
18     @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
19     private Long id;
20     @NotEmpty
21     @Size(min = 4, max = 40)
22     private String nom;
23     @Temporal(TemporalType.DATE)
24     @DateTimeFormat(pattern = "yyyy-MM-dd")
25     private Date dateNaissance;
26     private boolean malade;
27     @DecimalMin("100")
28     private int score;
29 }

```

### ➤ Interfaces DAO basées sur Spring Data

```

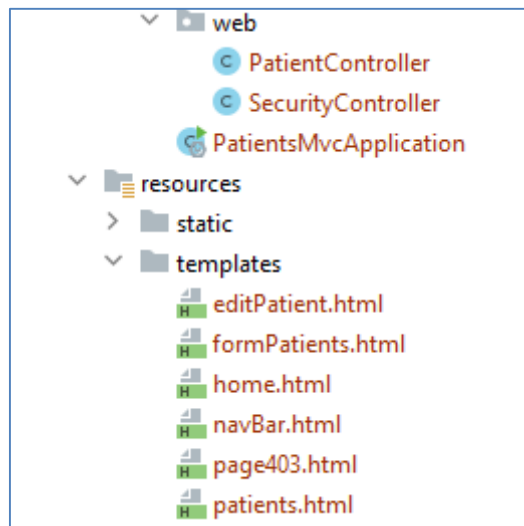
8
9 @Repository
10 public interface PatientRepository extends JpaRepository<Patient, Long> {
11     Page<Patient> findByNomContains(String kw, Pageable pageable);
12 }

```

## ➤ Test de la couche DAO

id	date_naissance	malade	score	nom
1	2022-05-27	1	129	Ahlam
2	2022-05-27	1	129	Salma
3	2022-05-27	0	129	Amjad
4	2022-05-27	0	129	Mohammed

## ➤ Couche Web



## ➤ interface

Patient Home Patients - ikram

Liste des patients

Key word

ID	Nom	Date	Malade	Score	Actions
1	Ahlam	2022-05-27	true	129	<input type="button" value="Delete"/> <input type="button" value="Update"/>
2	Salma	2022-05-27	true	129	<input type="button" value="Delete"/> <input type="button" value="Update"/>
3	Amjad	2022-05-27	false	129	<input type="button" value="Delete"/> <input type="button" value="Update"/>
4	Mohammed	2022-05-27	false	129	<input type="button" value="Delete"/> <input type="button" value="Update"/>
5	Ahlam	2022-05-27	false	129	<input type="button" value="Delete"/> <input type="button" value="Update"/>

Id :

Nom :

Date Naissance :

☒ Malade

Score :

## • Contrôleur

```

PatientController.java
19  @AllArgsConstructor
20  public class PatientController
21  {
22      private PatientRepository patientRepository;
23      @GetMapping(path = "/user/index")
24      @
25      public String patientsFunction(Model model,
26                                  @RequestParam(name = "page", defaultValue = "0") int page,
27                                  @RequestParam(name = "size", defaultValue = "5") int size,
28                                  @RequestParam(name = "keyword", defaultValue = "") String keyword){
29          Page<Patient> pagePatients = patientRepository.findByNomContains(keyword, PageRequest.of(page, size));
30          model.addAttribute( attributeName: "listePatients", pagePatients.getContent());
31          model.addAttribute( attributeName: "pagesNumber", new int[pagePatients.getTotalPages()]);
32          model.addAttribute( attributeName: "currentPage", page);
33          model.addAttribute( attributeName: "keyword", keyword);
34          //return une vue qui s'appelle patients.html
35          return "patients";
36      }
37
38      @GetMapping("/admin/delete")
39      public String deleteFunction(Long id,
40                                  @RequestParam(defaultValue = "") String keyword,
41                                  @RequestParam(defaultValue = "0") int page){
42          patientRepository.deleteById(id);
43          return "redirect:/user/index?page="+page+"&keyword="+keyword;
44      }

```

```

45  @GetMapping("/") public String home() { return "home"; }
46
47  // Rendu coté client retourner une liste Json des patients
48  // @RestController
49  @GetMapping("/user/patients")
50  @ResponseBody //obligatoire
51  public List<Patient> patientList(){
52      //DispatcherServlet comprendre qu'il s'agit pas d'une vue(String) et il va convertir on format Json
53      return patientRepository.findAll();
54  }
55
56  @GetMapping("/admin/formPatients")
57  public String formPatients(Model model){
58      model.addAttribute( attributeName: "patientform", new Patient());
59      return "formPatients";
60  }
61
62  //utiliser pour ajouter et pour faire la mise à jour
63  //si l'id est null ==> ajouter
64  //sinon ==> modifier
65  @PostMapping(path = "/admin/save")
66  @
67  public String save(Model model, @Valid Patient patient, BindingResult bindingResult,
68                      @RequestParam(defaultValue = "") String keyword,
69                      @RequestParam(defaultValue = "0") int page){
70      //valeur par défaut @RequestParam
71

```

```

72     if(bindingResult.hasErrors())
73     {if(patient.getId()==null) return "formPatients";
74       else return "editPatient";}
75     patientRepository.save(patient);
76     return "redirect:/user/index?page="+page+"&keyword="+keyword;
77   }
78
79   @GetMapping(path = "/admin/editPatient")
80   public String editPatient(Model model, Long id, String keyword, int page){
81     Patient patient = patientRepository.findById(id).orElse( other: null);
82     if(patient==null) throw new RuntimeException("Patient introuvable");
83     model.addAttribute( attributeName: "patientEdit", patient);
84     model.addAttribute( attributeName: "page", page);
85     model.addAttribute( attributeName: "keyword", keyword);
86     return "editPatient";}
87
88   }

```

## • Utilisation des Layouts :

Templates Généralement toutes les page d'une application web partagent le même contenu html (Header, footer, menus, etc..) Pour éviter des faire des copies coller dans toutes les pages, il est important de définir une page template (**navBar.html**) qui définit toutes les parties fixes de toutes les pages (header, footer, menus, etc...) et déclarer les sections qui changeront de contenu en fonction de chaque page.

```

<!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">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <link rel="stylesheet" type="text/css" href="/webjars/bootstrap/5.1.3/css/bootstrap.min.css">
  <!-- Add icon library -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

  <script src="webjars/bootstrap/5.1.3/js/bootstrap.bundle.js"></script>
</head>
<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand" th:href="@{/user/index}">Patient</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown" ar
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDropdown">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" th:href="@{/user/index}">Home</a>

```

## • Vue : patients.html

```

patients.html x
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org"
3     xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
4     xmlns:sec="http://www.thymeleaf.org/extras/spring-security"
5     layout:decorate="navBar">
6 <head>
7     <meta charset="UTF-8">
8     <title>Liste Patients</title>
9 </head>
10 <body>
11 <div layout:fragment="content1">
12 <div class="container mt-4">
13 <div class="card">
14 <div class="card-header">Liste des patients</div>
15 <div class="card-body">
16 <form method="get" th:action="@{/user/index}" class="form-control mb-4">
17 <label class="col-form-label">Key word</label>
18 <input type="text" name="keyword" th:value="${keyword}" style="...">
19 <button type="submit" class="btn btn-primary">Chercher</button>
20 </form>
21 <table class="table">
22 <thead>
23 <tr>

```

## • Vue : editPatient.html

```

editPatient.html x
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org"
3     xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
4     layout:decorate="navBar">
5 <head>
6 <meta charset="UTF-8">
7 <title>Edit Patient</title>
8 </head>
9 <body>
10 <div layout:fragment="content1">
11 <div class="card col-md-6 offset-3 mt-3" style="...">
12 <form method="post" th:action="@{/admin/save(page=${page}, keyword=${keyword})}">
13 <div>
14 <label class="col-form-label" for="id">Id : </label>
15 <input id="id" class="form-control" type="text" name="id" th:value="${patientEdit.id}" readonly>
16 </div>
17 <div>
18 <label class="col-form-label" for="nom">Nom : </label>
19 <input id="nom" class="form-control" type="text" name="nom" th:value="${patientEdit.nom}">
20 <span class="text-danger" th:errors="${patientEdit.nom}"></span>
21 </div>
22 </div>
23 </div>

```

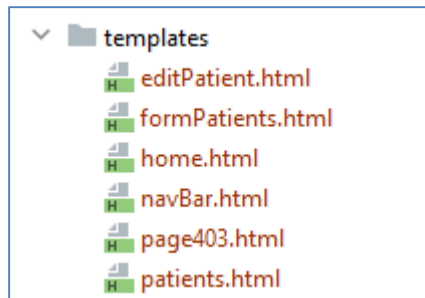
## • Vue : formPatient.html

```

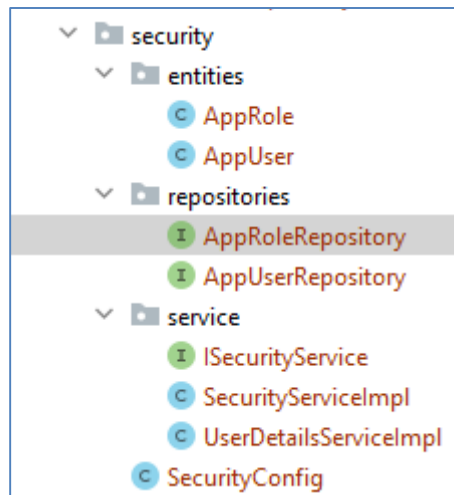
formPatients.html x
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org"
3     xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
4     layout:decorate="navBar">
5 <head>
6 <meta charset="UTF-8">
7 <title>Form Patients</title>
8 </head>
9 <body>
10 <div layout:fragment="content1">
11 <div class="card col-md-6 offset-3 mt-3" style="...">
12 <form method="post" th:action="@{/admin/save}">
13 <div>
14 <label class="col-form-label" for="nom">Nom : </label>
15 <input id="nom" class="form-control" type="text" name="nom" th:value="${patientform.nom}">
16 <span class="text-danger" th:errors="${patientform.nom}"></span>
17 </div>
18 <div>
19 <label class="col-form-label">Date Naissance : </label>
20 <input class="form-control" type="date" name="dateNaissance" th:value="${patientform.dateNaissance}">
21 <span class="text-danger" th:errors="${patientform.dateNaissance}"></span>
22 </div>
23 <div class="form-check mt-3">
24 <input type="checkbox" class="form-check-input" name="malade" th:checked="${patientform.malade}">

```

- Autres vues



## ➤ SPRING SECURITY



- SecurityConfig.java

```

14  @Configuration
15  @EnableWebSecurity //va etre instancié en premier
16  public class SecurityConfig extends WebSecurityConfigurerAdapter
17  {
18      @Autowired
19      private DataSource dataSource ;
20      @Autowired
21      private UserDetailsServiceImpl userDetailsService;
22      @Autowired
23      private PasswordEncoder passwordEncoder;
24      //pour spécifier les users
25      @Override
26      protected void configure(AuthenticationManagerBuilder auth) throws Exception
27      {
28          /*
29          String encodePWD = passwordEncoder.encode("1234");
30          System.out.println("=====> "+encodePWD);
31          //stocker dans la mémoire
32          auth.inMemoryAuthentication().withUser("user1")
33              .password(passwordEncoder.encode("1111")).roles("USER")
34              .and().withUser("user2").password(encodePWD).roles("USER");

```



```

38      /*
39      auth.jdbcAuthentication()
40          .dataSource(dataSource)
41          .usersByUsernameQuery("select username as principal, password as credentials, active from users where username=?")
42          .authoritiesByUsernameQuery("select username as principal, role as role from users_roles where username=?")
43          .rolePrefix("ROLE_")
44          .passwordEncoder(passwordEncoder);
45
46      */
47      auth.userDetailsService(userDetailsService);
48  }
49
50  //pour spécifier les droit d'accès de chaque user
51  @Override
52  @protected void configure(HttpSecurity http) throws Exception
53  {
54      http.formLogin(); //je veux utiliser un formulaire d'authentification fournit par Spring Security
55      //http.formLogin().loginPage("/login"); ma propre formulaire (créer méthode dans controller)
56
57      http.authorizeRequests().antMatchers("/").permitAll();
58      http.authorizeRequests().antMatchers("/admin/**").hasAuthority("ADMIN");
59      http.authorizeRequests().antMatchers("/user/**").hasAuthority("USER");
60      http.authorizeRequests().antMatchers("/webjars/**").permitAll();
61
62      http.authorizeRequests().anyRequest().authenticated();
63
64      http.authorizeRequests().anyRequest().authenticated();
65      // http.authorizeHttpRequests().anyRequest().authenticated(); //toutes les requetes http nécessitent une authentification
66
67      http.exceptionHandling().accessDeniedPage("/403"); // page 403
68  }

```

## ➤ Users

id	active	password	username
050ad1f4-3dc5-43e7-979a-7b4383bc2228	1	\$2a\$10\$6ucChF79tyVmxzh10VxGS.CexmMLUBHypWi.9f42V0...	Akram
595243a7-626f-43b5-a427-a2a2045ea5c9	1	\$2a\$10\$NqD.WCB4ccU0HdCZN5ROee03bOVbwhdka8rA/N3V4/0...	ikram
ad8a15e4-430b-49fc-8c85-232fa49a753b	1	\$2a\$10\$UIQ9HPcfQCnKRpmFWt1O4LOegTJ9SIKDv0TsayqKY...	Hassan

## ➤ Roles

role_id	description	role_name
1		ADMIN
2		USER

## ➤ Role\_Users

app_roles_role_id	users_id
1	595243a7-626f-43b5-a427-a2a2045ea5c9
2	050ad1f4-3dc5-43e7-979a-7b4383bc2228
2	595243a7-626f-43b5-a427-a2a2045ea5c9
2	ad8a15e4-430b-49fc-8c85-232fa49a753b

## • Interfaces de Spring Security de l'authentification

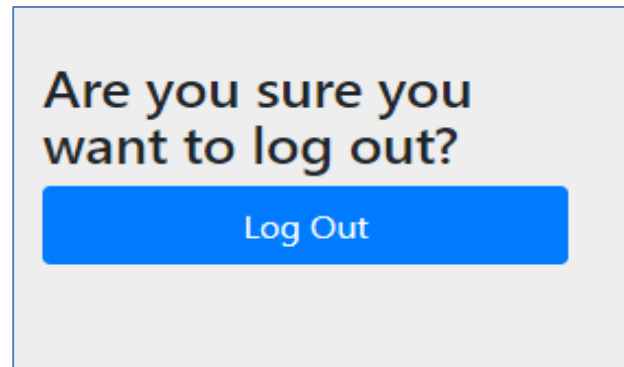
### ➤ Sign in

Please sign in



Sign in

➤ Log out



➤ Structure finale de projet :

