

2021-2022

**Département Mathématiques et Informatique
GLSID 2 – S4**

Architecture JEE et Middlewares

Rapport TP
Spring MVC Thymeleaf

Préparé par : Fatima Zahra HASBI
Encadré par : Mr Mohamed YOUSSEFI

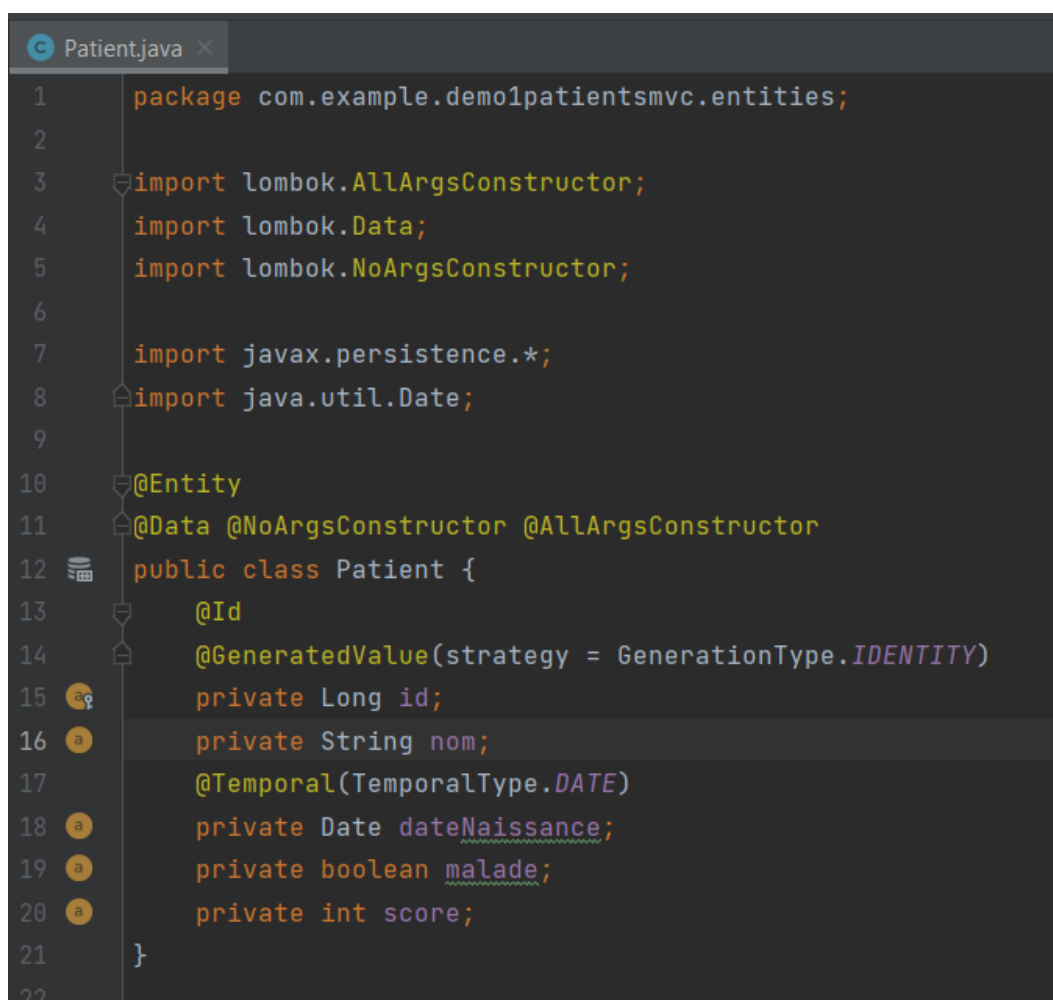
Introduction

Ce tp consiste à créer 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

Réalisation

Dans le dossier entities on crée la classe patient



```
1 package com.example.demo1patientsmvc.entities;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Data;
5 import lombok.NoArgsConstructor;
6
7 import javax.persistence.*;
8 import java.util.Date;
9
10 @Entity
11 @Data @NoArgsConstructor @AllArgsConstructor
12 public class Patient {
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private Long id;
16     private String nom;
17     @Temporal(TemporalType.DATE)
18     private Date dateNaissance;
19     private boolean malade;
20     private int score;
21 }
22
```

Dans le dossier repositories on crée l'interface PatientRepository

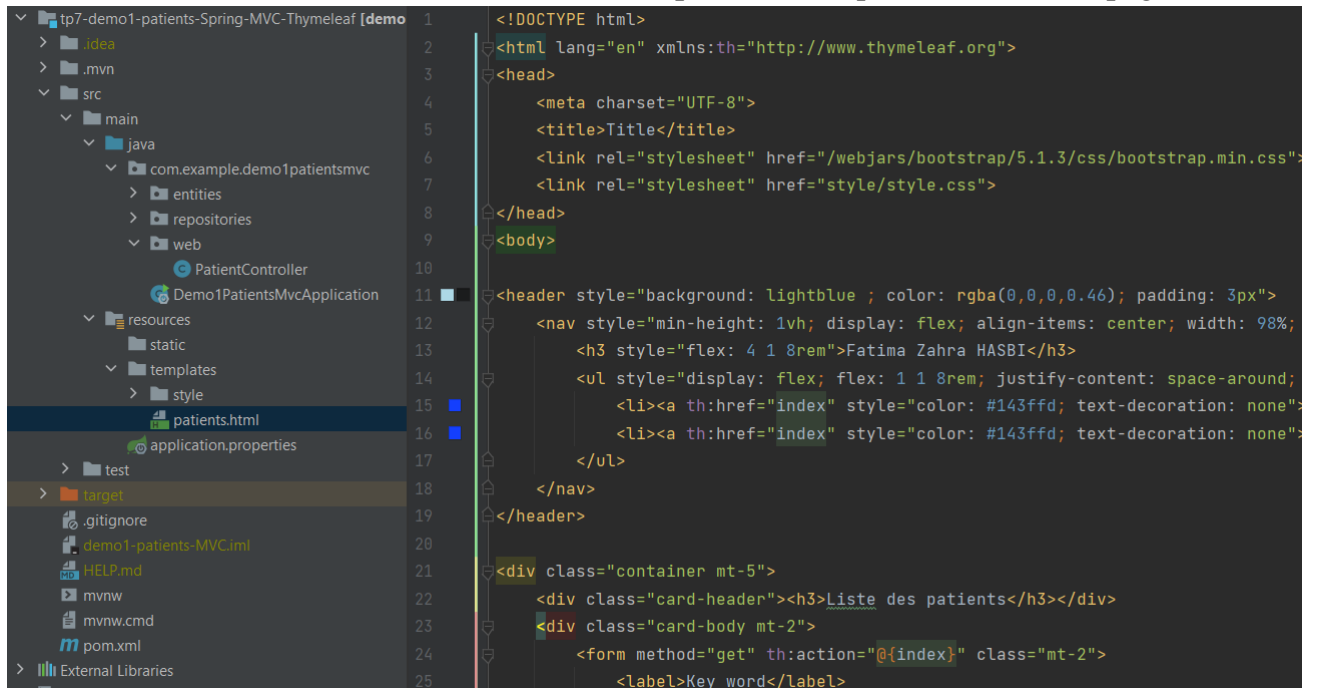
```
1 package com.example.demo1patientsmvc.repositories;
2
3 import com.example.demo1patientsmvc.entities.Patient;
4 import org.springframework.data.domain.Page;
5 import org.springframework.data.domain.Pageable;
6 import org.springframework.data.jpa.repository.JpaRepository;
7 import org.springframework.data.jpa.repository.Query;
8 import org.springframework.data.repository.query.Param;
9
10 import java.util.Date;
11 import java.util.List;
12
13 public interface PatientRepository extends JpaRepository<Patient,Long> {
14     public List<Patient> findByMalade(Boolean m);
15     Page<Patient> findByMalade(boolean m , Pageable pageable);
16     List<Patient> findByMaladeAndScoreLessThan(boolean m , int score);
17     List<Patient> findByMaladeIsFalseAndScoreLessThan(int score);
18     List<Patient> findByNomContains(String kw, Pageable pageable);
19     @Query("select p from Patient p where p.dateNaissance between :x and :y or p.nom like :z")
20     public List<Patient> FindPatients(@Param("x") Date d1, @Param("y") Date d2, @Param("z") String nom);
21 }
22
```

Dans le dossier web on crée le controleur de la page web PatientController

```
1 package com.example.demo1patientsmvc.web;
2
3 import com.example.demo1patientsmvc.entities.Patient;
4 import com.example.demo1patientsmvc.repositories.PatientRepository;
5 import lombok.AllArgsConstructor;
6 import org.springframework.data.domain.Page;
7 import org.springframework.data.domain.PageRequest;
8 import org.springframework.stereotype.Controller;
9 import org.springframework.ui.Model;
10 import org.springframework.web.bind.annotation.GetMapping;
11 import org.springframework.web.bind.annotation.RequestParam;
12
13 import java.util.ArrayList;
14 import java.util.List;
15
16 @Controller
17 @AllArgsConstructor
18 public class PatientController {
19     private PatientRepository patientRepository;
20     @GetMapping(path = "/index")
21     @RequestMapping
22     public String patients(Model model,
23                           @RequestParam(name = "page", defaultValue = "0") int page,
24                           @RequestParam(name = "size", defaultValue = "5") int size,
25                           @RequestParam(name = "keyword", defaultValue = "") String keyword){
26         Page<Patient> pagePatients=patientRepository.findAll(PageRequest.of(page, size));
27         List<Patient> patientList=new ArrayList<>();
28         model.addAttribute( attributeName: "patientList", pagePatients.getContent());
29         model.addAttribute( attributeName: "pages", new int[pagePatients.getTotalPages()]);
30         model.addAttribute( attributeName: "currentPage", page);
31         model.addAttribute( attributeName: "keyword", keyword);
32         return "patients";
33     }
34 }

```

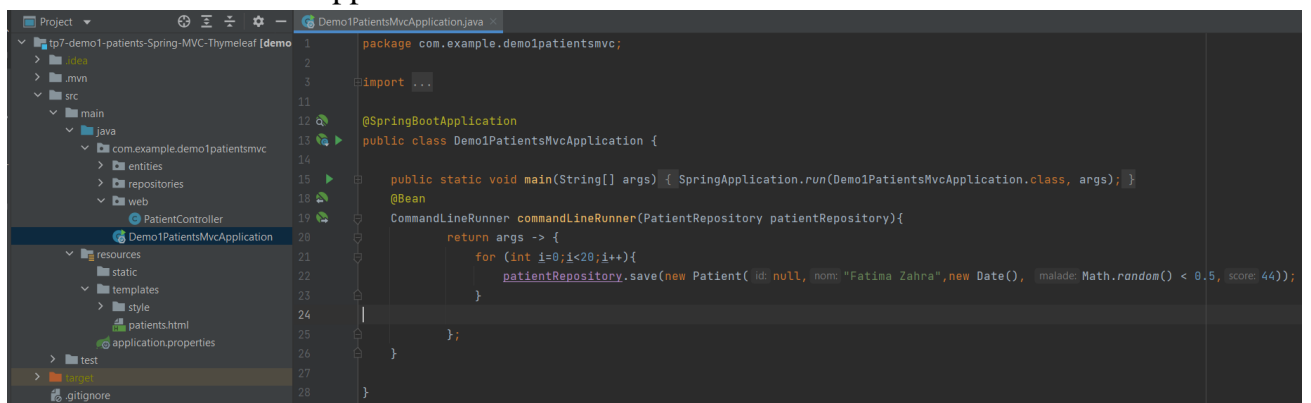
Sous le dossier ressource on crée un dossier template dans lequel on crée notre page html



The screenshot shows an IDE with a project structure on the left and an HTML file named `patients.html` open in the editor. The project structure includes a `src/main/resources/templates` directory. The `patients.html` file contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6   <link rel="stylesheet" href="/webjars/bootstrap/5.1.3/css/bootstrap.min.css">
7   <link rel="stylesheet" href="style/style.css">
8 </head>
9 <body>
10
11 <header style="background: lightblue ; color: rgba(0,0,0,0.46); padding: 3px">
12   <nav style="min-height: 1vh; display: flex; align-items: center; width: 98%;>
13     <h3 style="flex: 4 1 8rem">Fatima Zahra HASBI</h3>
14     <ul style="display: flex; flex: 1 1 8rem; justify-content: space-around;>
15       <li><a th:href="index" style="color: #143ffd; text-decoration: none;">
16       <li><a th:href="index" style="color: #143ffd; text-decoration: none;">
17     </ul>
18   </nav>
19 </header>
20
21 <div class="container mt-5">
22   <div class="card-header"><h3>Liste des patients</h3></div>
23   <div class="card-body mt-2">
24     <form method="get" th:action="@{index}" class="mt-2">
25       <label>Key word</label>
```

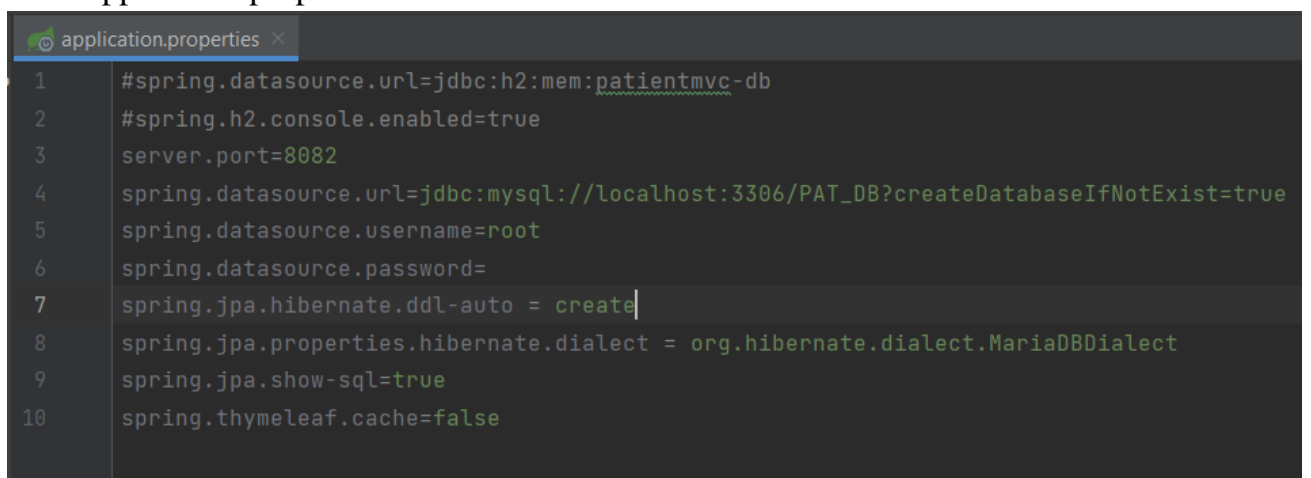
La fonction main de l'application



The screenshot shows an IDE with a project structure on the left and a Java file named `Demo1PatientsMvcApplication.java` open in the editor. The `main` method of the application is shown, which initializes the application and creates 20 patients.

```
1 package com.example.demo1patientsmvc;
2
3 import ...
4
5
6
7
8
9
10
11
12 @SpringBootApplication
13 public class Demo1PatientsMvcApplication {
14
15   public static void main(String[] args) { SpringApplication.run(Demo1PatientsMvcApplication.class, args); }
16
17   @Bean
18   CommandLineRunner commandLineRunner(PatientRepository patientRepository){
19     return args -> {
20       for (int i=0;i<20;i++){
21         patientRepository.save(new Patient( id: null, nom: "Fatima Zahra", new Date(), malade: Math.random() < 0.5, score: 44));
22       }
23     };
24   }
25 }
26
27
28 }
```

Dans application.properties



The screenshot shows an IDE with a project structure on the left and an `application.properties` file open in the editor. The file contains the following properties:

```
1 #spring.datasource.url=jdbc:h2:mem:patientmvc-db
2 #spring.h2.console.enabled=true
3 server.port=8082
4 spring.datasource.url=jdbc:mysql://localhost:3306/PAT_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
10 spring.thymeleaf.cache=false
```

Sur phpMyAdmin

Server: 127.0.0.1 » Database: pat_db » Table: patient

Showing rows 0 - 24 (119 total, Query took 0.0104 seconds.)

```
SELECT * FROM `patient`
```

Options: 1 > >> Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	date_naissance	malade	nom	score
<input type="checkbox"/>	1	2022-03-24	0	Fatima Zahra	44
<input type="checkbox"/>	2	2022-03-24	1	Fatima Zahra	44
<input type="checkbox"/>	3	2022-03-24	1	Fatima Zahra	44
<input type="checkbox"/>	4	2022-03-24	1	Fatima Zahra	44
<input type="checkbox"/>	6	2022-03-24	0	Fatima Zahra	44
<input type="checkbox"/>	7	2022-03-24	0	Fatima Zahra	44
<input type="checkbox"/>	8	2022-03-24	0	Fatima Zahra	44
<input type="checkbox"/>	9	2022-03-24	0	Fatima Zahra	44
<input type="checkbox"/>	10	2022-03-24	1	Fatima Zahra	44
<input type="checkbox"/>	11	2022-03-24	1	Fatima Zahra	44
<input type="checkbox"/>	12	2022-03-24	1	Fatima Zahra	44
<input type="checkbox"/>	13	2022-03-24	1	Fatima Zahra	44
<input type="checkbox"/>	14	2022-03-24	1	Fatima Zahra	44
<input type="checkbox"/>	15	2022-03-24	1	Fatima Zahra	44
<input type="checkbox"/>	16	2022-03-24	0	Fatima Zahra	44
<input type="checkbox"/>	17	2022-03-24	1	Fatima Zahra	44

Sur localhost/8082/

localhost:8082/index

Fatima Zahra HASBI

Home Les patients

Liste des patients

Key word

ID	Nom	Date	Malade	Score	
1	Fatima Zahra	2022-03-26	true	44	<input type="button" value="Delete"/>
2	Fatima Zahra	2022-03-26	false	44	<input type="button" value="Delete"/>
3	Fatima Zahra	2022-03-26	true	44	<input type="button" value="Delete"/>
4	Fatima Zahra	2022-03-26	true	44	<input type="button" value="Delete"/>
5	Fatima Zahra	2022-03-26	false	44	<input type="button" value="Delete"/>

0 1 2 3