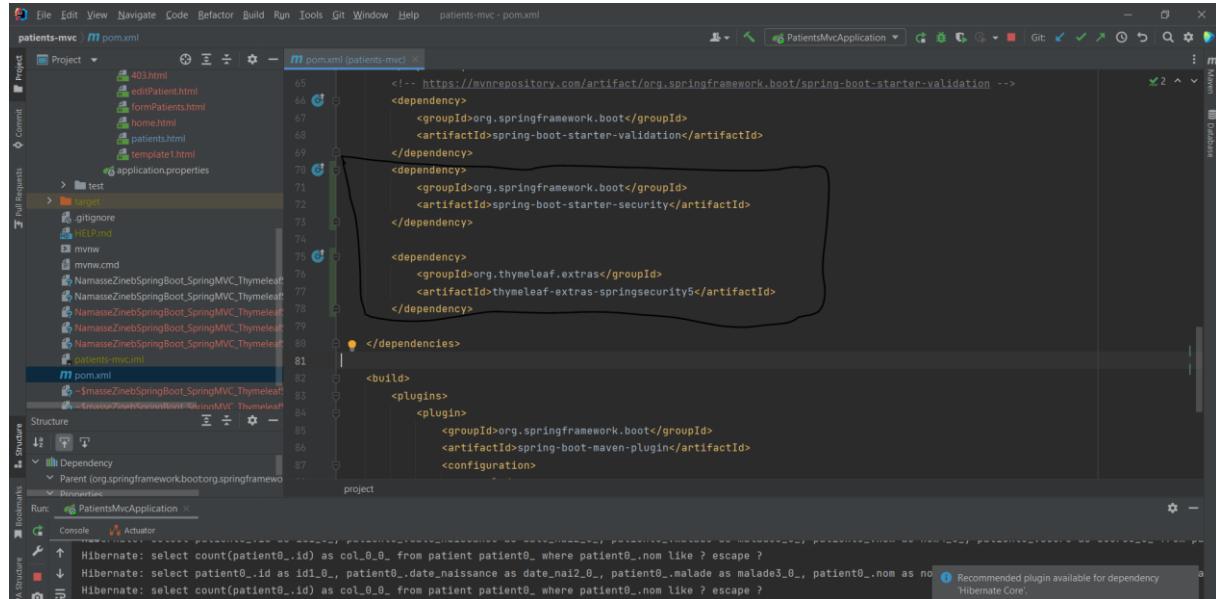


Spring Boot Spring MVC Thymeleaf Spring Data IntelliJ

(Partie 3)

pom.xml

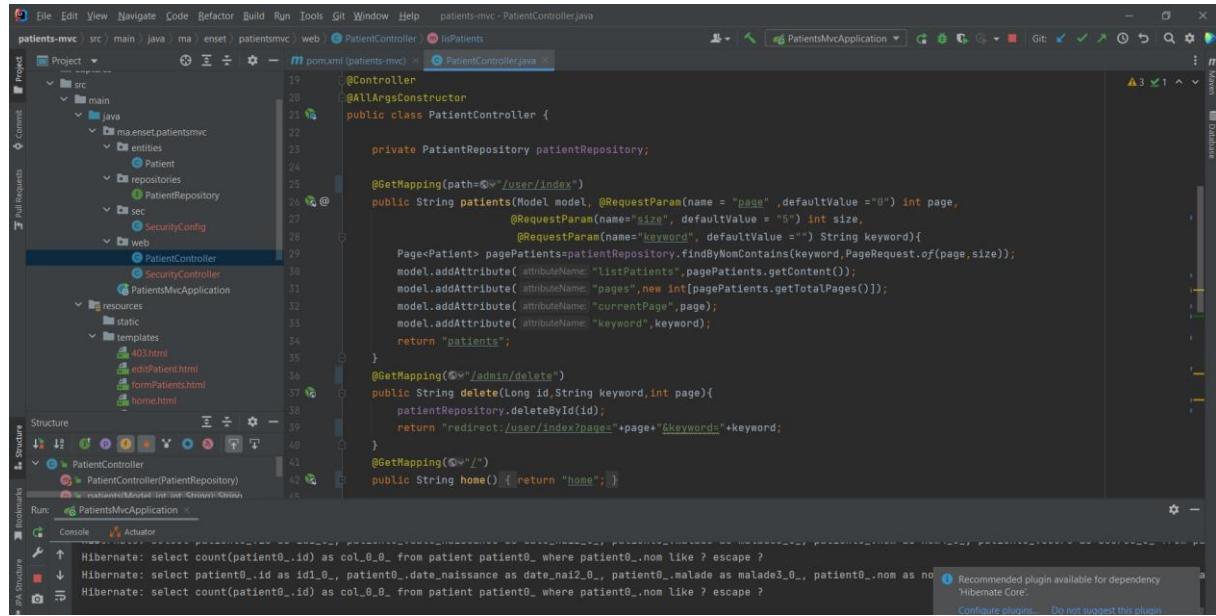


The screenshot shows the IntelliJ IDEA interface with the pom.xml file open. A callout box highlights the following dependencies:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-validation</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
    <groupId>org.thymeleaf.extras</groupId>
    <artifactId>thymeleaf-extras-springsecurity5</artifactId>
</dependency>
```

Below the dependencies, the build section includes the spring-boot-maven-plugin.

PatientController.java



The screenshot shows the IntelliJ IDEA interface with the PatientController.java file open. The code defines a PatientController with methods for listing patients, deleting a patient, and displaying the home page.

```
@Controller
@AutowireConstructor
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 "patients";
    }

    @GetMapping("/admin/delete")
    public String delete(Long id, String keyword, int page) {
        patientRepository.deleteById(id);
        return "redirect:/user/index?page=" + page + "&keyword=" + keyword;
    }

    @GetMapping("/")
    public String home() { return "home"; }
}
```

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Tree:** Shows the project structure under "patients-mvc".
- Editor:** Displays the code for `PatientController.java`. The code includes methods for `home`, `lisPatients`, `formPatients`, `save`, and `editPatient`.
- Toolbars and Status Bar:** Standard IntelliJ IDEA toolbars and status bar.

```
public String home() { return "home"; }

@GetMapping("/user/patients")
@ResponseBody
public List<Patient> lisPatients() { return patientRepository.findAll(); }

@GetMapping("/admin/formPatients")
public String formPatients(Model model){
    model.addAttribute("patient", new Patient());
    return "formPatients";
}

@PostMapping(path = @Value("/admin/save"))
public String save(Model model, @Valid Patient patient, BindingResult bindingResult,
                   @RequestParam(defaultValue = "0") int page,
                   @RequestParam(defaultValue = "") String keyword){
    if(bindingResult.hasErrors()) return "formPatients";
    patientRepository.save(patient);
    return "redirect:/user/index?page="+page+"&keyword="+keyword;
}

@GetMapping("/admin/editPatient")
public String editPatient(Model model, Long id, String keyword, int page){
    Patient patient=patientRepository.findById(id).orElse(null);
    if(patient==null) throw new RuntimeException("Patient introuvable");
}
```

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Tree:** Shows the project structure under "patients-mvc".
- Panels:** "Structure" panel is visible on the left.
- Status Bar:** Shows "Hibernate: select count(patient0_.id) as col_0_0_ from patient patient0_ where patient0_.nom like ? escape ?" and other database logs.
- Bottom Bar:** Shows "Recommended plugin available for dependency 'Hibernate Core...'".

```
public String formPatients(Model model){
    model.addAttribute("patient", new Patient());
    return "formPatients";
}

@PostMapping(path = @Value"/admin/save")
public String save(Model model, @Valid Patient patient, BindingResult bindingResult,
    @RequestParam(defaultValue = "") int page,
    @RequestParam(defaultValue = "") String keyword){
    if(bindingResult.hasErrors()) return "formPatients";
    patientRepository.save(patient);
    return "redirect:/user/index?page=" + page + "&keyword=" + keyword;
}

@GetMapping(@Value"/admin/editPatient")
public String editPatient(Model model, Long id, String keyword, int page){
    Patient patient = patientRepository.findById(id).orElse(throw new RuntimeException("Patient introuvable"));
    model.addAttribute("patient", patient);
    model.addAttribute("keyword", keyword);
    model.addAttribute("page", page);
    return "editPatient";
}
```

SecurityController.java

```

package ma.enset.patientsmvc.web;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

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

```

The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays the `SecurityController.java` file. The controller has a single method `notAuthorized()` annotated with `@GetMapping("/403")`. The code editor also shows the `pom.xml` and `PatientController.java` files. The bottom panel shows the run configuration for `PatientsMvcApplication`.

SecurityConfig.java

```

package ma.enset.patientsmvc.sec;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import javax.sql.DataSource;

@Configuration
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {
    @Autowired
    private DataSource dataSource;
    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        PasswordEncoder passwordEncoder=passwordEncoder();
        /...
        String encoder=DBUtil.getPasswordEncoder("1234");
    }
}

```

The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The code editor displays the `SecurityConfig.java` file. It is annotated with `@Configuration` and `@EnableWebSecurity`. It extends `WebSecurityConfigurerAdapter` and overrides the `configure` method to set up authentication using a password encoder. The code editor also shows the `pom.xml` and `PatientController.java` files. The bottom panel shows the run configuration for `PatientsMvcApplication`.

```
String encoderPWD=passwordEncoder.encode("1234");
System.out.println(encoderPWD);
auth.inMemoryAuthentication().withUser("user1").password(encoderPWD).roles("USER");
auth.inMemoryAuthentication().withUser("user2").password(passwordEncoder.encode("1111")).roles("USER");
auth.inMemoryAuthentication().withUser("admin").password(passwordEncoder.encode("2345")).roles("USER","ADMIN");
*/
auth.jdbcAuthentication()
    .dataSource(dataSource)
    .usersByUsernameQuery("select username as principal, password as credentials, active from users where username=?")
    .authoritiesByUsernameQuery("select username as principal, role as role from users_roles where username=?")
    .rolePrefix("ROLE_")
    .passwordEncoder(passwordEncoder);
```

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

```
.usersByUsernameQuery("select username as principal, password as credentials, active from users where username=?")
    .authoritiesByUsernameQuery("select username as principal, role as role from users_roles where username=?")
    .rolePrefix("ROLE_")
    .passwordEncoder(passwordEncoder);
```

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

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

403.html

The screenshot shows the IntelliJ IDEA interface with the project 'patients-mvc' open. The code editor displays the '403.html' template file. The code is as follows:

```
<!DOCTYPE html>


The project structure on the left shows files like PatientRepository.java, SecurityConfig.java, PatientController.java, SecurityController.java, and various HTML templates such as 403.html, editPatient.html, formPatients.html, home.html, patients.html, and template1.html.


```

editPatient.html

The screenshot shows the IntelliJ IDEA interface with the project 'patients-mvc' open. The code editor displays the 'editPatient.html' template file. The code is as follows:

```
<!DOCTYPE html>


The project structure on the left shows files like PatientRepository.java, SecurityConfig.java, PatientController.java, SecurityController.java, and various HTML templates such as 403.html, editPatient.html, formPatients.html, home.html, patients.html, and template1.html.


```

```

<span class="text-danger" th:errors="${patient.nom}">></span>
<div>
    <label>Date de naissance</label>
    <input class="form-control" type="date" name="dateNaissance" th:value="${patient.dateNaissance}">
    <span class="text-danger" th:errors="${patient.dateNaissance}">></span>
</div>
<div>
    <label>Malade</label>
    <input type="checkbox" name="malade" th:value="${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}">
    <span class="text-danger" th:errors="${patient.score}">></span>
</div>
<button type="submit" class="btn btn-primary">Save</button>
</div>
</body>
</html>

```

Run: PatientsMvcApplication

Console: Actuator

Hibernate: select count(patient0_.id) as col_0_0 from patient patient0_ where patient0_.nom like ? escape ?
Hibernate: select patient0_.id as id1_0_, patient0_.date_naissance as date_na12_0_, patient0_.malade as malade3_0_, patient0_.nom as nom0_ from patient patient0_ where patient0_.nom like ? escape ?
Hibernate: select count(patient0_.id) as col_0_0 from patient patient0_ where patient0_.nom like ? escape ?

Externally added files can be added to Git // View Files // Always Add // Don't Ask Again (3 minutes ago)

formPatients.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>Form Patients</title>
</head>
<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 de 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>
                </div>
            </form>
        </div>
    </div>
</body>

```

Run: PatientsMvcApplication

Console: Actuator

Hibernate: select count(patient0_.id) as col_0_0 from patient patient0_ where patient0_.nom like ? escape ?
Hibernate: select patient0_.id as id1_0_, patient0_.date_naissance as date_na12_0_, patient0_.malade as malade3_0_, patient0_.nom as nom0_ from patient patient0_ where patient0_.nom like ? escape ?
Hibernate: select count(patient0_.id) as col_0_0 from patient patient0_ where patient0_.nom like ? escape ?

Externally added files can be added to Git // View Files // Always Add // Don't Ask Again (3 minutes ago)

```
<div>
    <label>Date de 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:value="${patient.malade}">
    <span class="text-danger" th:errors="${patient.malade}"></span>
</div>
<div>
    <label>Score</label>
    <input class="form-control" type="text" name="score" th:value="${patient.score}">
    <span class="text-danger" th:errors="${patient.score}"></span>
</div>
<button type="submit" class="btn btn-primary">Save</button>
</form>
</div>
</body>
</html>
```

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">
<head>
    <meta charset="UTF-8">
    <title>Index</title>
</head>
<body>
    <div layout:fragment="content"></div>
</body>
</html>
```

patients.html

The screenshot shows an IDE interface with two tabs open: PatientController.java and patients.html.

PatientController.java:

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org"
3   xmlns:layout="http://www.ultrad.net.nz/thymeleaf/layout"
4     layout:decorate="template1"
5     xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
6   <head>
7     <meta charset="UTF-8">
8     <title>Title</title>
9   </head>
10  <body>
11    <div layout:fragment="content">
12      <div class="card">
13        <div class="card-header">Liste des patients</div>
14        <div class="card-body">
15          <form method="get" th:action="@{/user/index}">
16            <label>Key word</label>
17            <input type="text" name="keyword" th:value="${keyword}">
18            <button type="submit" class="btn btn-primary">Chercher</button>
19          </form>
20          <table class="table">
21            <thead>
22              <tr>
23                <th>ID</th><th>Nom</th><th>Date de Naissance</th><th>Malade</th><th>Score</th>
24              </tr>
25            </thead>
26            <tbody>
27              <tr th:each="p : ${listPatients}">
28                <td th:text="${p.id}"></td>
29                <td th:text="${p.nom}"></td>
30                <td th:text="${p.dateNaissance}"></td>
31                <td th:text="${p.malade}"></td>
32                <td th:text="${p.score}"></td>
33                <td sec:authorize="hasRole('ADMIN')">
34                  <a onclick="return confirm('Etes-vous sûr(e) ?')" class="btn btn-danger" th:href="@{/admin/delete(id=${p.id}, keyword=${keyword}, page=${currentPage})}">Delete</a>
35                  <a class="btn btn-success" th:href="@{/admin/editPatient(id=${p.id}, keyword=${keyword}, page=${currentPage})}">Edit</a>
36                </td>
37              </tr>
38            </tbody>
39          </table>
40        </div>
41      </div>
42    </div>
43  </body>
44</html>
```

patients.html:

```
<div layout:fragment="content">
  <div class="card">
    <div class="card-header">Liste des patients</div>
    <div class="card-body">
      <form method="get" th:action="@{/user/index}">
        <label>Key word</label>
        <input type="text" name="keyword" th:value="${keyword}">
        <button type="submit" class="btn btn-primary">Chercher</button>
      </form>
      <table class="table">
        <thead>
          <tr>
            <th>ID</th><th>Nom</th><th>Date de Naissance</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="hasRole('ADMIN')">
              <a onclick="return confirm('Etes-vous sûr(e) ?')" class="btn btn-danger" th:href="@{/admin/delete(id=${p.id}, keyword=${keyword}, page=${currentPage})}">Delete</a>
              <a class="btn btn-success" th:href="@{/admin/editPatient(id=${p.id}, keyword=${keyword}, page=${currentPage})}">Edit</a>
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
```

The screenshot shows an IDE interface with two tabs open: PatientController.java and patients.html.

PatientController.java:

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org"
3   xmlns:layout="http://www.ultrad.net.nz/thymeleaf/layout"
4     layout:decorate="template1"
5     xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
6   <head>
7     <meta charset="UTF-8">
8     <title>Title</title>
9   </head>
10  <body>
11    <div layout:fragment="content">
12      <div class="card">
13        <div class="card-header">Liste des patients</div>
14        <div class="card-body">
15          <form method="get" th:action="@{/user/index}">
16            <label>Key word</label>
17            <input type="text" name="keyword" th:value="${keyword}">
18            <button type="submit" class="btn btn-primary">Chercher</button>
19          </form>
20          <table class="table">
21            <thead>
22              <tr>
23                <th>ID</th><th>Nom</th><th>Date de Naissance</th><th>Malade</th><th>Score</th>
24              </tr>
25            </thead>
26            <tbody>
27              <tr th:each="p : ${listPatients}">
28                <td th:text="${p.id}"></td>
29                <td th:text="${p.nom}"></td>
30                <td th:text="${p.dateNaissance}"></td>
31                <td th:text="${p.malade}"></td>
32                <td th:text="${p.score}"></td>
33                <td sec:authorize="hasRole('ADMIN')">
34                  <a onclick="return confirm('Etes-vous sûr(e) ?')" class="btn btn-danger" th:href="@{/admin/delete(id=${p.id}, keyword=${keyword}, page=${currentPage})}">Delete</a>
35                  <a class="btn btn-success" th:href="@{/admin/editPatient(id=${p.id}, keyword=${keyword}, page=${currentPage})}">Edit</a>
36                </td>
37              </tr>
38            </tbody>
39          </table>
40        </div>
41      </div>
42    </div>
43  </body>
44</html>
```

patients.html:

```
<div layout:fragment="content">
  <div class="card">
    <div class="card-header">Liste des patients</div>
    <div class="card-body">
      <form method="get" th:action="@{/user/index}">
        <label>Key word</label>
        <input type="text" name="keyword" th:value="${keyword}">
        <button type="submit" class="btn btn-primary">Chercher</button>
      </form>
      <table class="table">
        <thead>
          <tr>
            <th>ID</th><th>Nom</th><th>Date de Naissance</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="hasRole('ADMIN')">
              <a onclick="return confirm('Etes-vous sûr(e) ?')" class="btn btn-danger" th:href="@{/admin/delete(id=${p.id}, keyword=${keyword}, page=${currentPage})}">Delete</a>
              <a class="btn btn-success" th:href="@{/admin/editPatient(id=${p.id}, keyword=${keyword}, page=${currentPage})}">Edit</a>
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
```

The screenshot shows the IntelliJ IDEA interface with the 'patients-mvc' project open. The 'patients.html' file is selected in the left navigation bar. The code editor displays the following HTML and Thymeleaf code:

```
</td>
<td sec:authorize="hasRole('ADMIN')">
    <a class="btn btn-success" th:href="@{/admin/editPatient(id=${p.id}, keyword=${keyword}, page=${currentPage})}">Edit</a>
</td>
</tr>
</tbody>
</table>


- 

```

template1.html

The screenshot shows the IntelliJ IDEA interface with the 'patients-mvc' project open. The 'template1.html' file is selected in the left navigation bar. The code editor displays the following HTML and Thymeleaf code:

```
<!DOCTYPE html>

```

The screenshot shows an IDE interface with the following details:

- Project Structure:** The project is named "patients-mvc". It contains a "PatientRepository" interface, a "SecurityConfig" class, and a "PatientController" class.
- Resources:** The "resources" folder contains "static" and "templates" subfolders. The "templates" folder includes files like "403.html", "editPatient.html", "formPatients.html", "home.html", "patients.html", and "template1.html".
- Code Editor:** The "template1.html" file is open. It contains HTML and JSP code for a navigation bar. A dropdown menu is shown, with one item disabled ("Disabled") and another item ("Nouveau Patient") which is annotated with `sec:authorize='hasRole('ADMIN')'`.
- Console:** The console shows Hibernate SQL queries being executed.
- Toolbars and Menus:** Standard IDE menus like File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, Git, Window, Help are visible.
- Status Bar:** The status bar at the bottom right shows "3366 CRLF, UTF-8, 4 spaces, master".

This screenshot is nearly identical to the first one, showing the same IDE environment and code editor content. The main difference is the state of the dropdown menu in the navigation bar, which now has both items enabled ("Nouveau Patient" and "Logout").

Base de données

The screenshot shows the phpMyAdmin interface for the 'pat_db' database. The 'roles' table is currently selected, showing 2 rows. Below the table list, there is a 'Nouvelle table' (New table) form with 'Nom:' set to 'nouveau_table' and 'Nombre de colonnes:' set to 4. The 'Exécuter' (Execute) button is visible.

Table roles:

The screenshot shows the 'roles' table in phpMyAdmin. The table has two rows: 'ADMIN' and 'USER'. The 'ADMIN' row has an edit icon. The 'USER' row has both edit and delete icons. The SQL query used to select the data is shown in the top left: 'SELECT * FROM `roles`'.

Table users

Table users

Structure:

```

CREATE TABLE `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `active` tinyint(1) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;

```

Data:

	username	password	active
1	admin	\$2a\$10\$guk1PxAcKezpYn0fqplMu02694RuL1o6.4FuRADcAw...	1
2	user1	\$2a\$10\$guk1PxAcKezpYn0fqplMu02694RuL1o6.4FuRADcAw...	1
3	user2	\$2a\$10\$guk1PxAcKezpYn0fqplMu02694RuL1o6.4FuRADcAw...	1

Table users_roles

Table users_roles

Structure:

```

CREATE TABLE `users_roles` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) NOT NULL,
  `role` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

```

Data:

	username	role
1	admin	ADMIN
2	user1	USER
3	user1	ADMIN
4	user1	USER
5	user2	USER

user2 (login) (rôle=user)

Please sign in

user2

....

Sign in

user2(index)

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "localhost:8082/user/index". The page displays a table of patient records with columns: ID, Nom, Date de Naissance, Malade, and Score. A search bar at the top allows filtering by "Key word". Below the table is a navigation menu with links: Home, Link, Link, Disabled, Patients, and a dropdown for "user2".

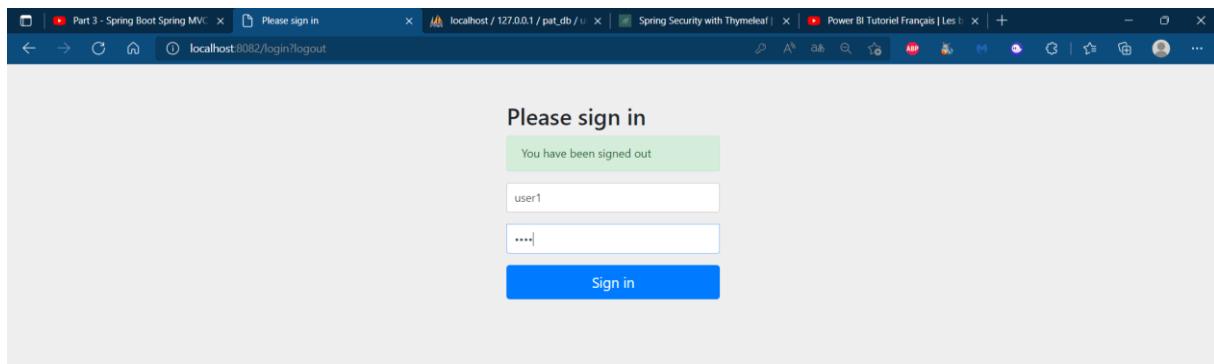
ID	Nom	Date de Naissance	Malade	Score
1	Yassine	2022-04-06	false	123
3	Yasmine	2022-04-06	true	65
4	Hanae	2022-04-06	false	32
5	Hassan	2022-04-06	false	12
6	Mohammed	2022-04-06	true	321

Below the table is a set of numbered buttons from 0 to 52, with the first few buttons (0, 1, 2, 3, 4) highlighted in blue.

user2(logout)

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "localhost:8082/logout". It displays a confirmation message: "Are you sure you want to log out?" with a "Log Out" button below it.

user1(login) (role=admin)



The screenshot shows a web browser window with multiple tabs open. The active tab is titled "localhost:8082/login?logout". It displays a "Please sign in" form with two input fields: "user1" and "....". Above the form, a green message box says "You have been signed out". A "Sign in" button is at the bottom of the form.

user2(index)

localhost:8082/user/index

Home Link Link Disabled Patients user1

Nouveau Patient
List Chercher Patient

Key word Chercher

ID	Nom	Date de Naissance	Malade	Score		
1	Yassine	2022-04-06	false	123	<button>Delete</button>	<button>Edit</button>
3	Yasmine	2022-04-06	true	65	<button>Delete</button>	<button>Edit</button>
4	Hanae	2022-04-06	false	32	<button>Delete</button>	<button>Edit</button>
5	Hassan	2022-04-06	false	12	<button>Delete</button>	<button>Edit</button>
6	Mohammed	2022-04-06	true	321	<button>Delete</button>	<button>Edit</button>
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	30	31	32	33	34
35	36	37	38	39	40	41
42	43	44	45	46	47	48
49	50	51	52			

admin(login)

localhost:8082/login?logout

Please sign in

You have been signed out

admin

.....

Sign in

admin(index)

localhost:8082/user/index

Home Link Link Disabled Patients admin

Nouveau Patient
List Chercher Patient

Key word Chercher

ID	Nom	Date de Naissance	Malade	Score		
1	Yassine	2022-04-06	false	123	<button>Delete</button>	<button>Edit</button>
3	Yasmine	2022-04-06	true	65	<button>Delete</button>	<button>Edit</button>
4	Hanae	2022-04-06	false	32	<button>Delete</button>	<button>Edit</button>
5	Hassan	2022-04-06	false	12	<button>Delete</button>	<button>Edit</button>
6	Mohammed	2022-04-06	true	321	<button>Delete</button>	<button>Edit</button>
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	30	31	32	33	34
35	36	37	38	39	40	41
42	43	44	45	46	47	48
49	50	51	52			