

Compte Rendu Contrôle

1. Introduction

Dans ce contrôle on présente une activité pratique avec ses quatre parties qui est une application web JEE destiné à la Gestion des patients.

2. Enonce

Ce projet consiste à la réalisation de quatre parties pratique qui ce présente comme ceci :

- Partie 1 : Rechercher les patients, pagination et suppression des patients.
- Partie 2 : Page template, Ajout des patients, validation des formulaires, édition et mise à jour des patients.
- Partie 3: Spring Security (Stratégies: InMemoryAuthentication ET JDBCAuthentication).
- Partie 4: Spring Security (Stratégie: UserDetailsService).

3. Réalisation

Pour réaliser ce projet y'en a deux classes :

- La Classe Patient :

```
package ma.emsi.patientsmvc.entities;

import ...
@Entity @AllArgsConstructor @NoArgsConstructor
@Data
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;
}
```

- La Classe Médecin :

```
package ma.emsi.patientsmvc.entities;

import ...
@Entity
@Data @AllArgsConstructor @NoArgsConstructor
public class Medecin {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nom;
    private String prenom;
    private String email;
    private String numero;
    private String specialite;
    @OneToMany(mappedBy = "medecin", fetch = FetchType.LAZY)
    @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
    private Collection<RendezVous> rendezVous;
}
```

✓ Code source

<https://github.com/NisrineR/jee>

✓ Capture d'écran

a. REPOSITORIES

- Patient REPOSITORY :

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

- Médecin REPOSITORY :

```

public interface MedecinRepository extends JpaRepository<Medecin, Long> {
    //Medecin findByNom(String nom);
    Page<Medecin> findByNomContains(String kw, Pageable pageable);
}

```

b. Partie CONTROLLER :

➤ Patient CONTROLLER :

```

@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 "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";
    }
}

```

➤ Médecin CONTROLLER :




































```
@Controller
@AllArgsConstructor
public class MedecinController {
    private MedecinRepository medecinRepository;

    @GetMapping(path = "/user/medecin")
    public String Medecin(Model model,
        @RequestParam(name = "page",defaultValue = "0") int page,
        @RequestParam(name = "size",defaultValue = "5")int size,
        @RequestParam(name = "keyword",defaultValue = "")String keyword
    ){
        Page<Medecin> pageMedecin=medecinRepository.findByNomContains(keyword, PageRequest.of(page, size));
        model.addAttribute( attributeName: "listmedecin",pageMedecin.getContent());
        model.addAttribute( attributeName: "pages",new int[pageMedecin.getTotalPages()]);
        model.addAttribute( attributeName: "currentPage",page);
        model.addAttribute( attributeName: "keyword",keyword);
        return "Medecin";
    }

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

    @GetMapping("/admin/FormMedecin")
    public String formMedecin(Model model){
        model.addAttribute( attributeName: "Medecin",new Medecin());
        return "formMedecin";
    }
}
```

✓ Les tables :

Table ▲	Action	Lignes ⓘ	Type	Interclassement	Taille	Perte
<input type="checkbox"/> app_role	★      	2	InnoDB	utf8mb4_general_ci	32,0 kio	-
<input type="checkbox"/> app_user	★      	2	InnoDB	utf8mb4_general_ci	32,0 kio	-
<input type="checkbox"/> app_user_app_roles	★      	3	InnoDB	utf8mb4_general_ci	48,0 kio	-
<input type="checkbox"/> consultation	★      	0	InnoDB	utf8mb4_general_ci	32,0 kio	-
<input type="checkbox"/> medecin	★      	1	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> patient	★      	2	InnoDB	utf8mb4_general_ci	16,0 kio	-

✓ Partie SECURITY :

a) Security Config :

```
@Configuration
@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {

    @Autowired
    private DataSource dataSource;

    @Autowired
    private UserDetailsService userDetailsService;

    @Autowired
    private PasswordEncoder passwordEncoder;

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception{
```

```
        auth.userDetailsService(userDetailsService);

    }

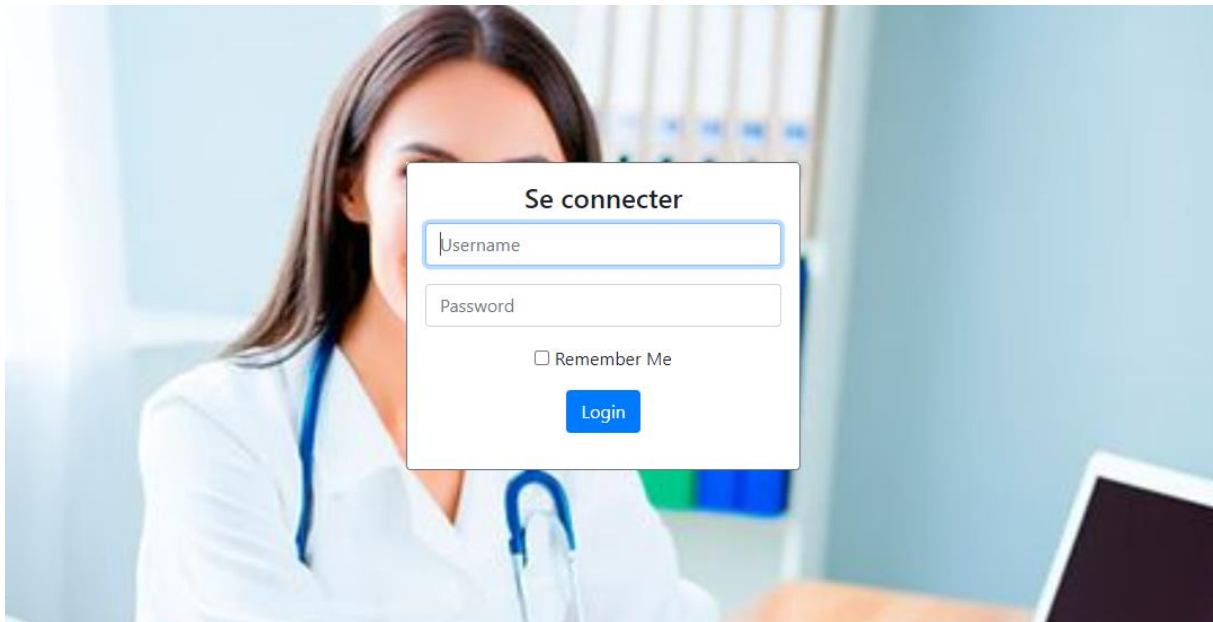
    @Override
    protected void configure(HttpSecurity http) throws Exception{
        http.formLogin();
        http.authorizeRequests().antMatchers("/*").permitAll();
        http.authorizeRequests().antMatchers("/admin/**").hasAuthority("ADMIN");
        http.authorizeRequests().antMatchers("/index/**").hasAuthority("USER");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

b) Security Controller :

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

✓ Exécution :

a. Login :



b. Nouveau Patient :

[Home](#) [Patients](#) [Medecin](#) nirine

Nom

Date Naissance

jj/mm/aaaa

Malade

☐

Score

0

Save

c) Chercher Patient :

[Home](#) [Patients](#) [Medecin](#) nirine

Liste des patients

Key word

Chercher

ID	Nom	Date	Malade	Score	Action
4	soukaina	2022-04-09	true	333	<div>Delete</div> <div>Edit</div>
20	amina	2022-05-13	false	124	<div>Delete</div> <div>Edit</div>
21	yasmine	2022-05-21	false	587	<div>Delete</div> <div>Edit</div>

0

Réaliser par :
Raihane Nisrine

EMSI ORANGER G21