

## DÉPARTEMENT MATHÉMATIQUES ET INFORMATIQUE

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

## TP 4 : Spring MVC, Spring Data JPA, Spring Security

**Réalisé Par:** Laila Sad Elbouyoud  
**Classe:** ii-BDCC 2

# Interface Graphique

## Page Login:

Please sign in

Username

Password

Sign in

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

Accueil Etudiants ▼

laila ▼

Liste des Etudiants

keyword

Chercher

ID	Nom	Prénom	Email	Date Naissance	Genre	Régle		
1	laila	saad	lailasd@gmail.com	2022-04-21	FEMININ	true	Supprimer	Modifier
2	sarah	alaoui	alaoui@gmail.com	2022-04-21	FEMININ	true	Supprimer	Modifier
3	imane	amine	imaneam@gmail.com	2022-04-21	FEMININ	false	Supprimer	Modifier
7	ahmed	ali	ahmedali@gmail.com	2022-04-21	MASCULIN	true	Supprimer	Modifier
8	laila	saad	lailasd@gmail.com	2022-04-21	FEMININ	true	Supprimer	Modifier

0

1

2

3

4

5

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

Accueil

Etudiants ▾

yassine ▾

Liste des Etudiants

keyword  Chercher

ID	Nom	Prénom	Email	Date Naissance	Genre	Régle
1	laila	saad	lailasd@gmail.com	2022-04-21	FEMININ	true
2	sarah	alaoui	alaoui@gmail.com	2022-04-21	FEMININ	true
3	imane	amine	imaneam@gmail.com	2022-04-21	FEMININ	false
7	ahmed	ali	ahmedali@gmail.com	2022-04-21	MASCULIN	true
8	laila	saad	lailasd@gmail.com	2022-04-21	FEMININ	true

0

1

2

3

4

5

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

## Page de déconnexion:

Are you sure you  
want to log out?

Log Out

## Page de Modification:

Accueil Etudiants laila

ID : 1

Nom

laila

Prénom

saad

Email

lailasd@gmail.com

Date Naissance

04/21/2022

Genre

FEMININ

Règle ☒

Soumettre

## Suppression:

- Message d'alerte:

localhost:8082 says

Etes vous surs?

OK

Cancel

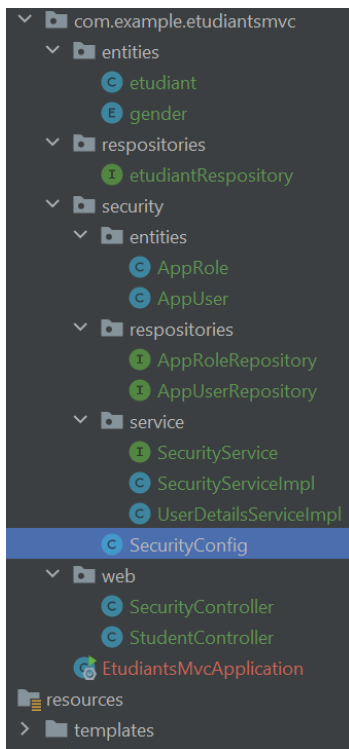
Chercher

n	Email	Date Naissance	Genre	Règle		
	lailasd@gmail.com	2022-04-21	FEMININ	true	Supprimer	Modifier
	alaoui@gmail.com	2022-04-21	FEMININ	true	Supprimer	Modifier
	imaneam@gmail.com	2022-04-21	FEMININ	false	Supprimer	Modifier
	ahmedali@gmail.com	2022-04-21	MASCULIN	true	Supprimer	Modifier
	lailasd@gmail.com	2022-04-21	FEMININ	true	Supprimer	Modifier

5

# Partie Code

## La structure du projet:



## Les entités:

### etudiant:

```
@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
public class etudiant {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @NotEmpty
    private String nom;
    @NotEmpty
    private String prenom;
    private String email;
    @Temporal(TemporalType.DATE)
    @DateTimeFormat(pattern = "yyyy-MM-dd")
    private Date dateNaissance;
    @Enumerated(EnumType.STRING)
    private gender genre;
    private boolean regle;
}
```

## Les énumérations:

### gender:

```
public enum gender {  
    FEMININ,  
    MASCULIN  
}
```

## Respositories:

### etudiantRespository:

```
public interface etudiantRespository extends JpaRepository<etudiant, Long> {  
    Page<etudiant> 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 pas");  
        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:

### studentController:

```
@Controller
@AllArgsConstructor
public class StudentController {
    private etudiantRespository etudiantRepository;

    @GetMapping(path = "/user/index")
    public String students(Model model,
        @RequestParam(name="page", defaultValue = "0") int page,
        @RequestParam(name="size", defaultValue = "5") int size,
        @RequestParam(name="keyword", defaultValue = "") String keyword){
        Page<etudiant> pageEtudiants = etudiantRepository.findByNomContains(keyword, PageRequest.of(page, size));
        model.addAttribute( attributeName: "listStudents", pageEtudiants.getContent());
        model.addAttribute( attributeName: "pages", new int[pageEtudiants.getTotalPages()]);
        model.addAttribute( attributeName: "currentPage", page);
        model.addAttribute( attributeName: "keyword", keyword);
        return "student";
    }
}
```

### SecurityController:

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

## EtudiantsMVCAApplication:

```
//@Bean
CommandLineRunner saveUsers(SecurityService securityService){
    return args -> {
        securityService.saveUser( username: "laila", password: "1234", rePassword: "1234");
        securityService.saveUser( username: "mohammed", password: "1234", rePassword: "1234");
        securityService.saveUser( username: "yassine", 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: "mohammed", roleName: "USER");
        securityService.addRoleToUser( username: "yassine", 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>
```

### editStudent.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 Etudiant</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="${student.id}"></label>
        <input class="form-control" type="hidden" name="id" th:value="${student.id}">
      </div>
      <div>
        <label for="nom">Nom</label>
        <input id="nom" class="form-control" type="text" name="nom" th:value="${student.nom}">
      </div>
    </form>
  </div>
</div>
```

formStudents.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="${student.nom}">
        <span class="text-danger" th:errors="${student.nom}"></span>
      </div>
      <div>
        <label for="prenom">Prénom</label>
        <input id="prenom" class="form-control" type="text" name="prenom" th:value="${student.prenom}">
        <span class="text-danger" th:errors="${student.prenom}"></span>
      </div>
      <div>
        <label for="email">Email</label>
        <input id="email" class="form-control" type="text" name="email" th:value="${student.email}">
        <span class="text-danger" th:errors="${student.email}"></span>
      </div>
      <div>
        <label>Date Naissance</label>
        <input class="form-control" type="date" name="dateNaissance" th:value="${student.dateNaissance}">
      </div>
    </form>
  </div>
</div>
</body>
```

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>
```

### student.html:

```
<div layout:fragment="content1">
  <div class="container mt-2">
    <div class="card">
      <div class="card-header">Liste des Etudiants</div>
      <div class="card-body">
        <form method="get" th:action="@{/user/index}">
          <label>keyword</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>Prénom</th>
              <th>Email</th>
              <th>Date Naissance</th>
              <th>Genre</th>
              <th>Régle</th>
            </tr>
          </thead>
        </table>
      </div>
    </div>
  </div>
```

### template1.html:

```
<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 dropdown">
        <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown">Etu
        <ul class="dropdown-menu">
          <li sec:authorize="hasAuthority('ADMIN')"><a class="dropdown-item" th:href="@{/admin
          <li><a class="dropdown-item" th:href="@{/user/index}">Chercher</a></li>
        </ul>
      </li>
    </ul>
    <ul class="navbar-nav">
      <li class="nav-item dropdown" sec:authorize="isAuthenticated()" >
        <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown">
          <span sec:authentication="name"></span>
        </a>
        <ul class="dropdown-menu">
```

## Application properties:

```
spring.datasource.url=jdbc:mysql://localhost:3306/db_students?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
```