

**Année : 2022/2023**

**Application web JEE, Gestion d'un hopital**

|  |  |
| --- | --- |
| Realisée par :  Ibtissam Elguerdi | Encadré par M. Youssfi Mohammed |

**Sommaire:**

**Sujet**

**Annotations et mots clés**

**Dependances**

**Structure de projet**

**Les Entités**

**Déployer le data source**

**Couche DAO avec Spring data**

**Repositories**

**Couche Web**

**Controleurs**

**Vues**

**Authentification SPRING SECURITY**

## **Sujet**

L’objectif principal consiste à concevoir et réaliser une application web dynamique avec le Framework Spring qui permet la gestion d’un hôpital. Les données sont stockées dans une base de données MySQL

L’application se compose de trois couches :

La couche DAO qui est basée sur Spring Data, JPA, Hibernate et JDBC.

La couche Métier

La couche Web basée sur MVC coté Serveur en utilisant Thymeleaf.

La sécurité est basée sur Spring Security

### Annotations et mots clés

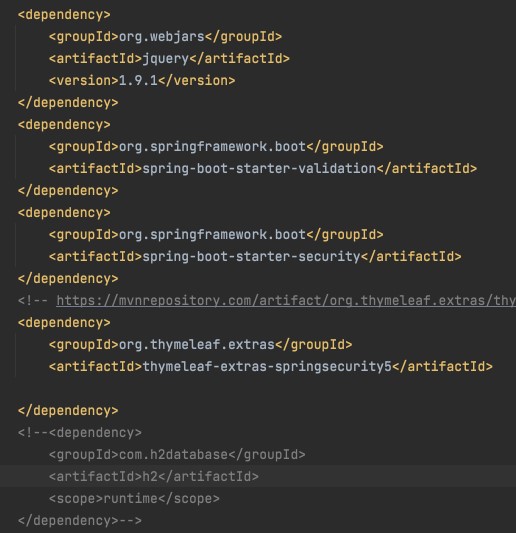
|  |  |  |  |
| --- | --- | --- | --- |
| Injection de dépendance | Permet d'implémenter le principe de l'inversion de contrôle   |  |  | | --- | --- | | Inversion de contrôle | => découpler les dépendances | |
| JDBC  (Java Data Base Connection) | API qui permet d’utiliser les bases de données relationnelles |
| Mapping Objet Relationnel    (ORM) | géré l’accès aux données |

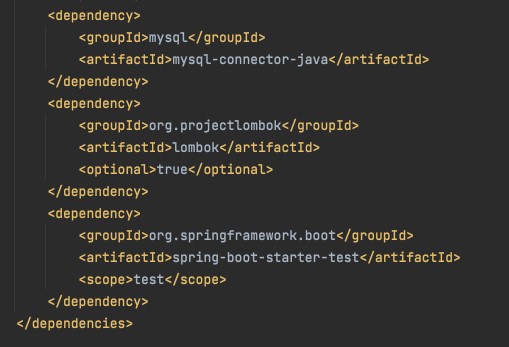
|  |  |
| --- | --- |
| Object-Relational Mapping |  |
| Hibernate | Un ORM qui implémente la spécification JPA |
| JPA (Java Persistance API) | Un API qui repose essentiellement sur l'utilisation des annotations |
| Spring Data | Un module de Spring qui facilite l’utilisation de JPA |
| Spring Data JPA | Fait l’ORM basé sur JPA |
| Lombok | Permet de générer les getters et les setters |
| Spring Web | Spring mvc |
| H2 Database | SGBD à mémoire  (les données sont perdus apres chaque redémarrage) |
| application.properties | fichier de configuration de l’application |
| Entities | Package qui contient les classes qui vont etre par la suite des tables dans la bd |

|  |  |
| --- | --- |
| @Data | annotation de Lombok  => gener les getters et setters et constructeurs |
| @ Entity | =>annotation essentiel pour une classe pour devenir une entité JPA |
| @ID | Primary key  =>annotation essentiel pour une classe pour devenir une entité JPA |
| repositories | Package qui contient des interfaces qui hérite de l’interface JpaRepository    => permet d’utiliser JPA |
| @Query | Annotation qui dit a SpringData comment interprète la fonction  =>Utilise HQL (Hibernate query language) |
| @Bean | Dit a Spring d’exécuter celle-ci au démarrage |
| BindingResult | informations sur les erreurs de validation |
| @Valid | Hibernate va utiliser les annotations de validations (comme @Min et @NotEmpty) avant d’exécuter les requêtes sql |
| @configuration | Dit a Spring que cette classe doit être instancier au premier lieu |

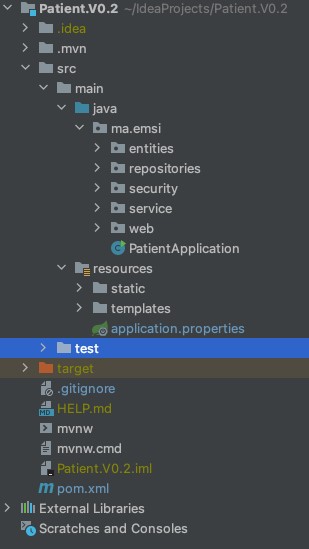
### Dependances



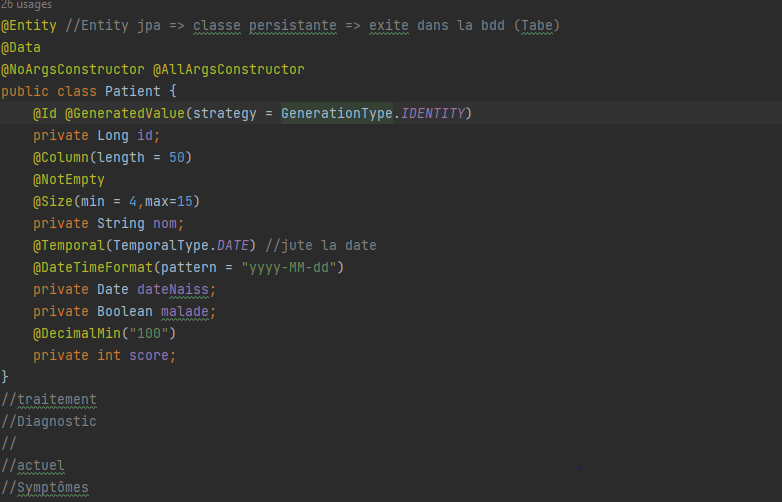




### Structure de projet



### Les Entités

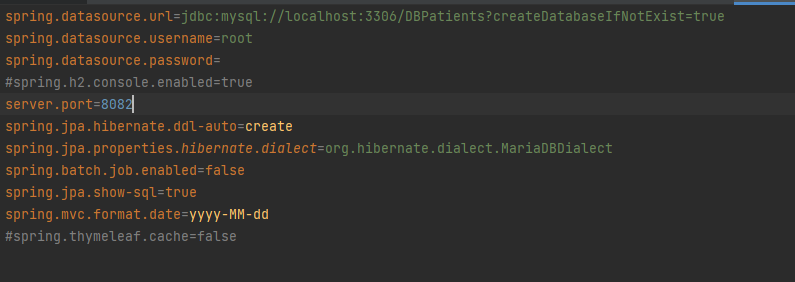


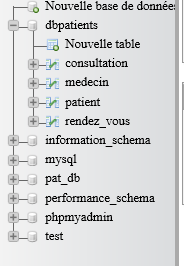




**Déployer le data source**

Le fichier de configuration ‘application.properties’





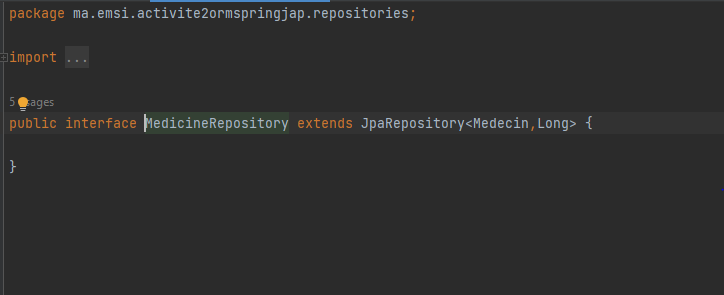
**Couche DAO avec Spring data**

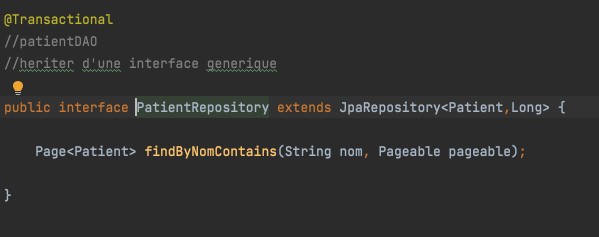
1. On a créer un package Repository
2. On a declaré pour chaque entité une interface EntityRepository qui hérite del’interface générique JpaRepository
3. On a declarer les signatures des méthodes en respectant les règles de l'écriturepour récupérer les données sans avoir besoin de les implémenter grâce à Spring

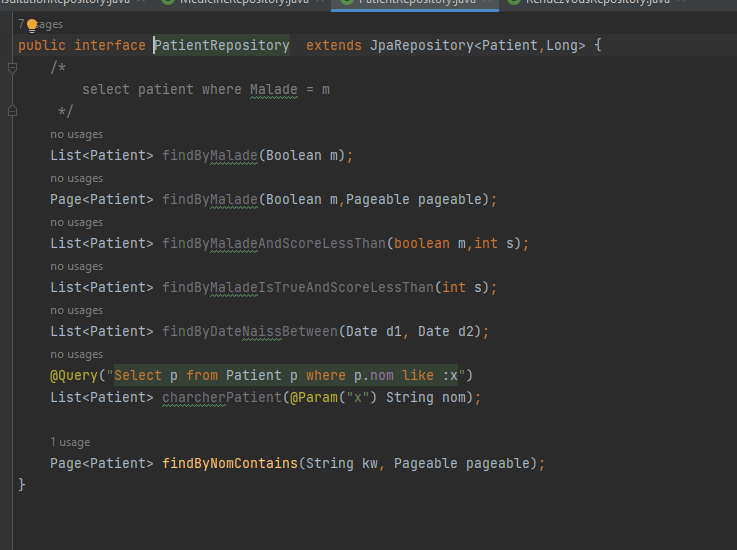
Boot

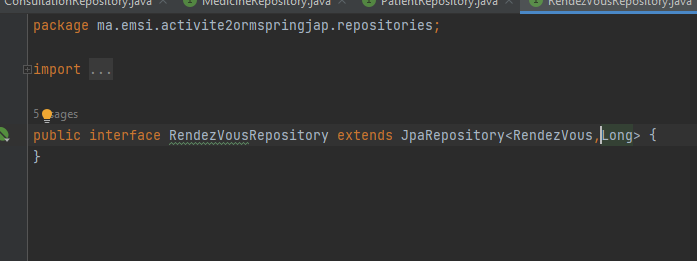
## Repositories

Les interfaces Repositories

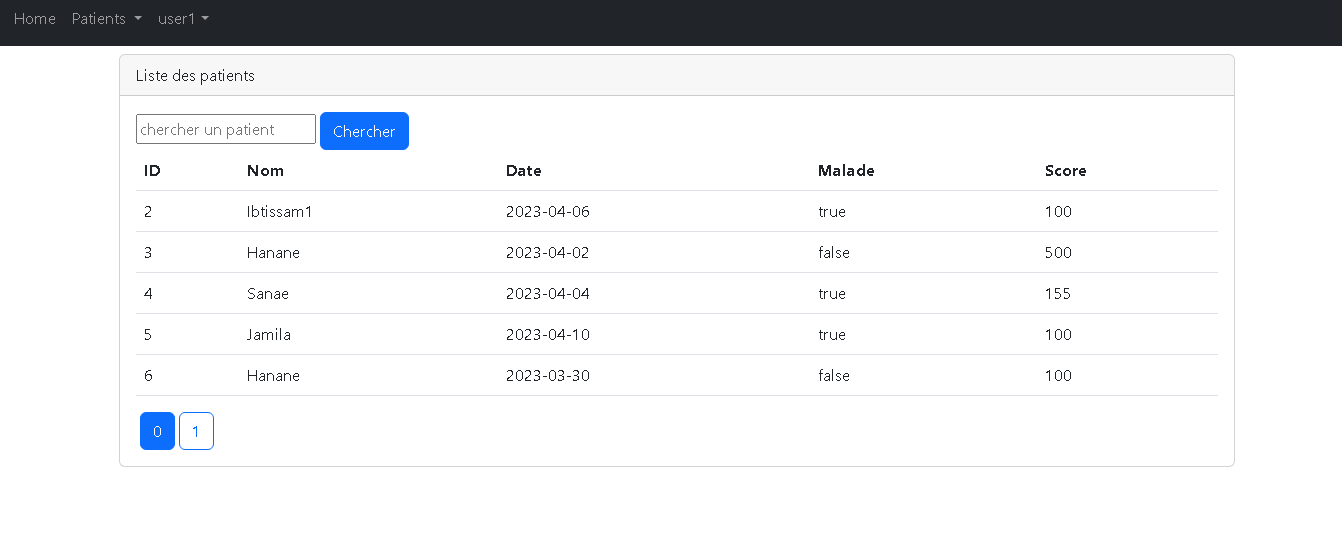


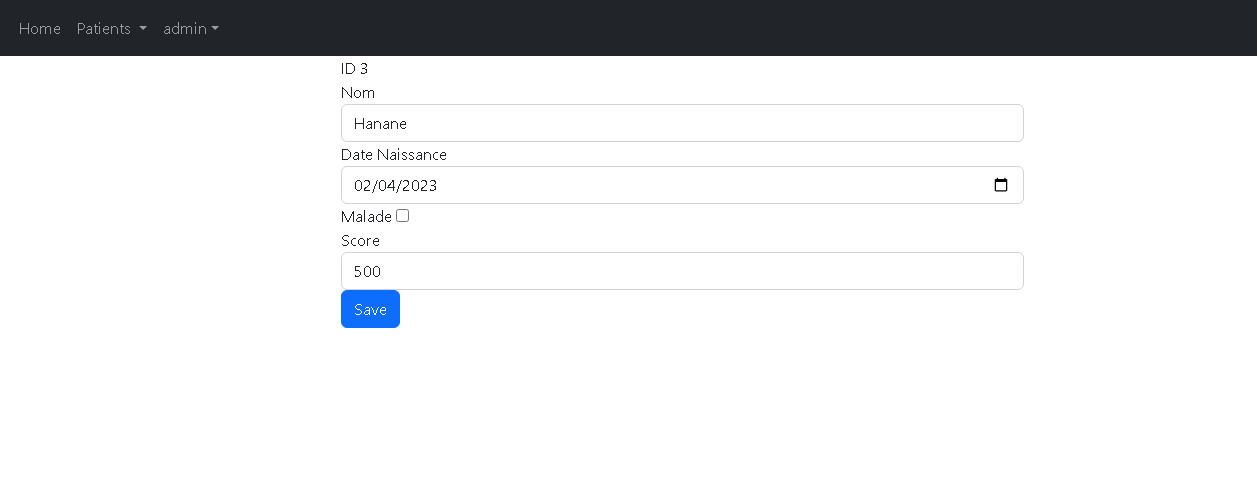






## Couche Web

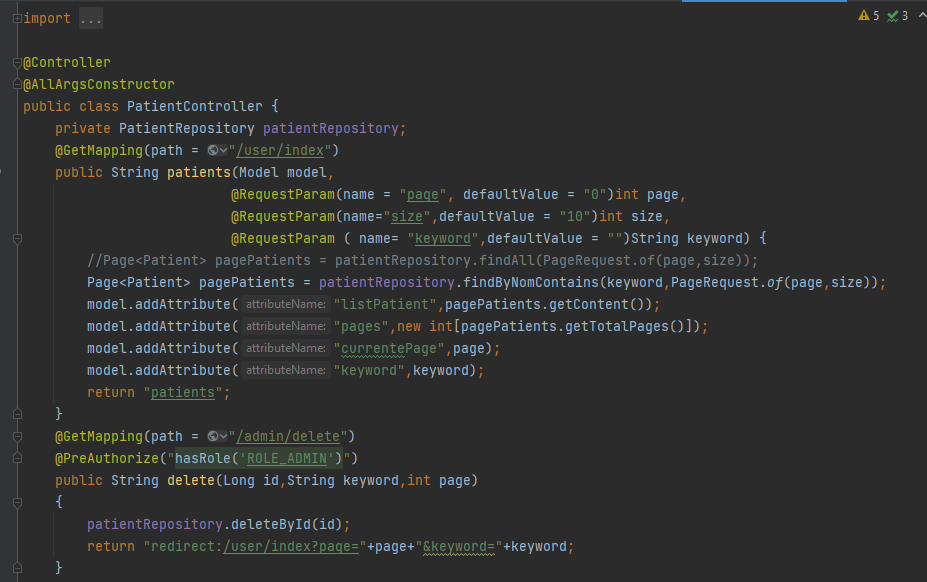




## Controleurs

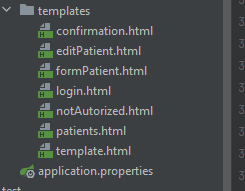
Une classe qui gère les requêtes http

La route ‘/user/index est liée à la méthode patients() elle va être appelée lorsqu’une requête de type GET est envoyée



## Vues

On a travaillée avec le moteur de template thymeleaf



**On ajout le dialect thyemeleaf**

****

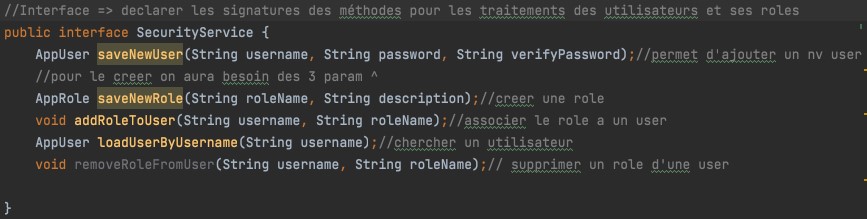
**Exemple d’une vue**



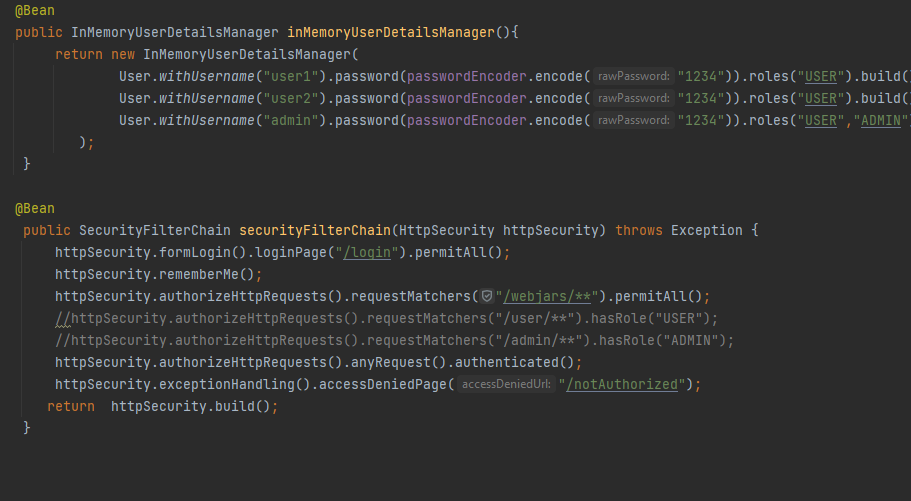
La methode save() reçoit les paramètres suivant: un objet de type Patient, quel page, sa taille et le keyword de la recherche

## Authentification SPRING SECURITY

Interface SecurityService qui contient la déclaration des signatures des méthodes qui concerne les utilisateurs et des rôles:



La méthode suivante prend comme paramètre HttpSecurity et va servir a spécifier les droits d’accès



Le résultat est un mot de passe difficile déchiffrer

