

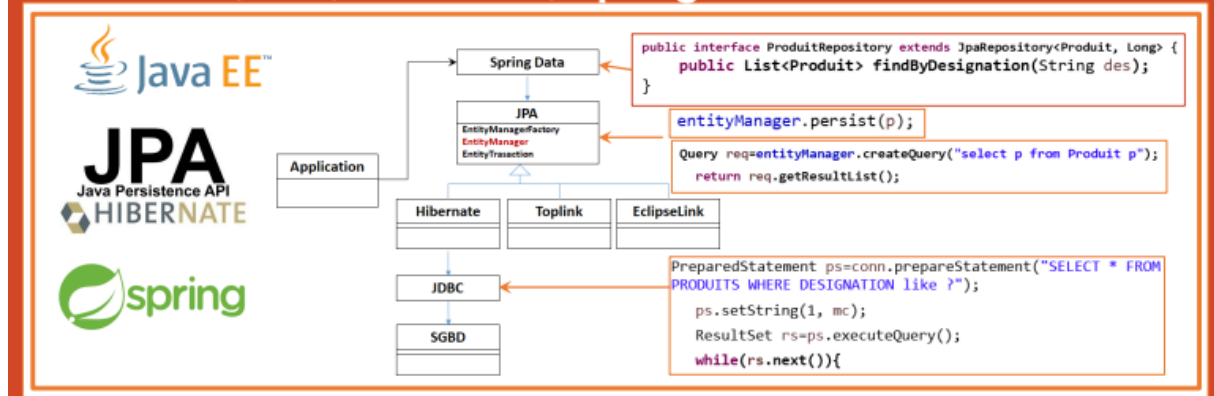
Activite2 :
Spring DATA



Réaliser par :
Mohamed amine
KHAMMOUR

Professeur :
Mr. Mohamed
YOUSSEFI

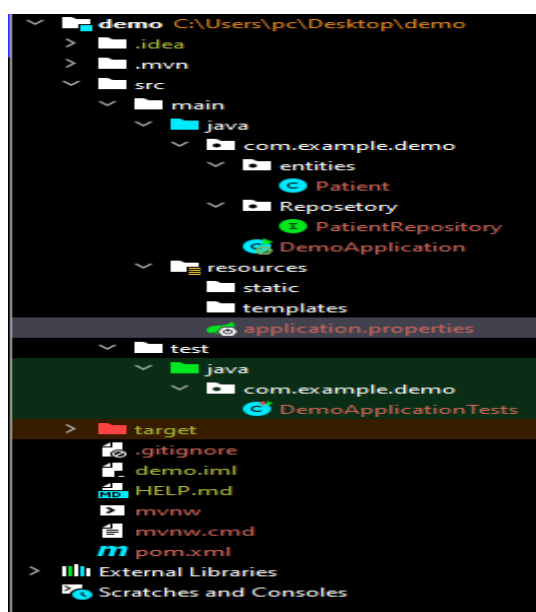
JEE : ORM, JPA, Hibernate, Spring Data



JPA, Hibernate, Spring Data

- Spring Data est un module de Spring qui a déjà créé des interfaces génériques et des implémentations génériques qui permettent de gérer les entités JPA.
- En utilisant Spring Data, vous n'aurez plus besoin de faire appel à l'objet EntityManager pour gérer la persistance. Spring Data le fait à votre place.
- Spring Data nous évite de créer les interfaces et les implémentations JPA de la couche DAO.
- Il suffit de créer une interface qui hérite de l'interface JpaRepository pour hériter toutes les méthodes classiques qui permettent de gérer les entités JPA.
- En cas de besoin, vous avez la possibilité d'ajouter d'autres méthodes en les déclarant à l'intérieur de l'interface JpaRepository, sans avoir besoin de les implémenter. Spring Data le fera à votre place.

Structure de projet Spring



La class patient

```
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.ToString;

import javax.persistence.*;
import java.util.Date;

@Entity
@Data @NoArgsConstructor @AllArgsConstructor
public class Patient {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Column(length = 50)
    private String nom;
    @Temporal(TemporalType.DATE)
    private Date date;
    private boolean malade;
    private int score;
}
```

L'interface PatientRepository

```
package com.example.demo.Repository;

import com.example.demo.entities.Patient;
import org.springframework.data.jpa.repository.JpaRepository;

public interface PatientRepository extends JpaRepository<Patient, Long> {

}
```

La classe DemoAplication

```
package com.example.demo;

import com.example.demo.Repository.PatientRepository;
import com.example.demo.entities.Patient;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.util.Date;
import java.util.List;

@SpringBootApplication
public class DemoApplication implements CommandLineRunner {
    @Autowired
    private PatientRepository patientRepository;

    public static void main(String[] args){SpringApplication.run(DemoApplication.class, args);}

    @Override
    public void run(String... args) throws Exception {
        for(int i=0;i<100;i++){
            patientRepository.save(new Patient(id: null, nom: "khalid", new Date(), i%2!=0?true:false, (int)(Math.random()*100)));

            patientRepository.save(new Patient(id: null, nom: "hassana", new Date(), i%2!=0?true:false, (int)(Math.random()*100)));
        }
        List<Patient> patients=patientRepository.findAll();
        patients.forEach(p->{
            System.out.println("=====");
        });
    }
}
```

Le fichier application.properties

```
spring.datasource.url=jdbc:h2:mem:patient-db
spring.h2.console.enabled=true
server.port=8082
```

L'exécution :

[illegible]

La base de données :

English Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:mem:patient-db

User Name: sa

Password:

Connect Test Connection

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM PATIENT

SELECT * FROM PATIENT;

ID	DATE	MALADE	NOM	SCORE
2	2022-03-04	FALSE	hassana	58
3	2022-03-04	TRUE	khalid	76
4	2022-03-04	TRUE	hassana	1
5	2022-03-04	FALSE	khalid	82
6	2022-03-04	FALSE	hassana	81
7	2022-03-04	TRUE	khalid	66
8	2022-03-04	TRUE	hassana	28
9	2022-03-04	FALSE	khalid	43
10	2022-03-04	FALSE	hassana	95
11	2022-03-04	TRUE	khalid	57
12	2022-03-04	TRUE	hassana	47
13	2022-03-04	FALSE	khalid	67
14	2022-03-04	FALSE	hassana	26
15	2022-03-04	TRUE	khalid	37