

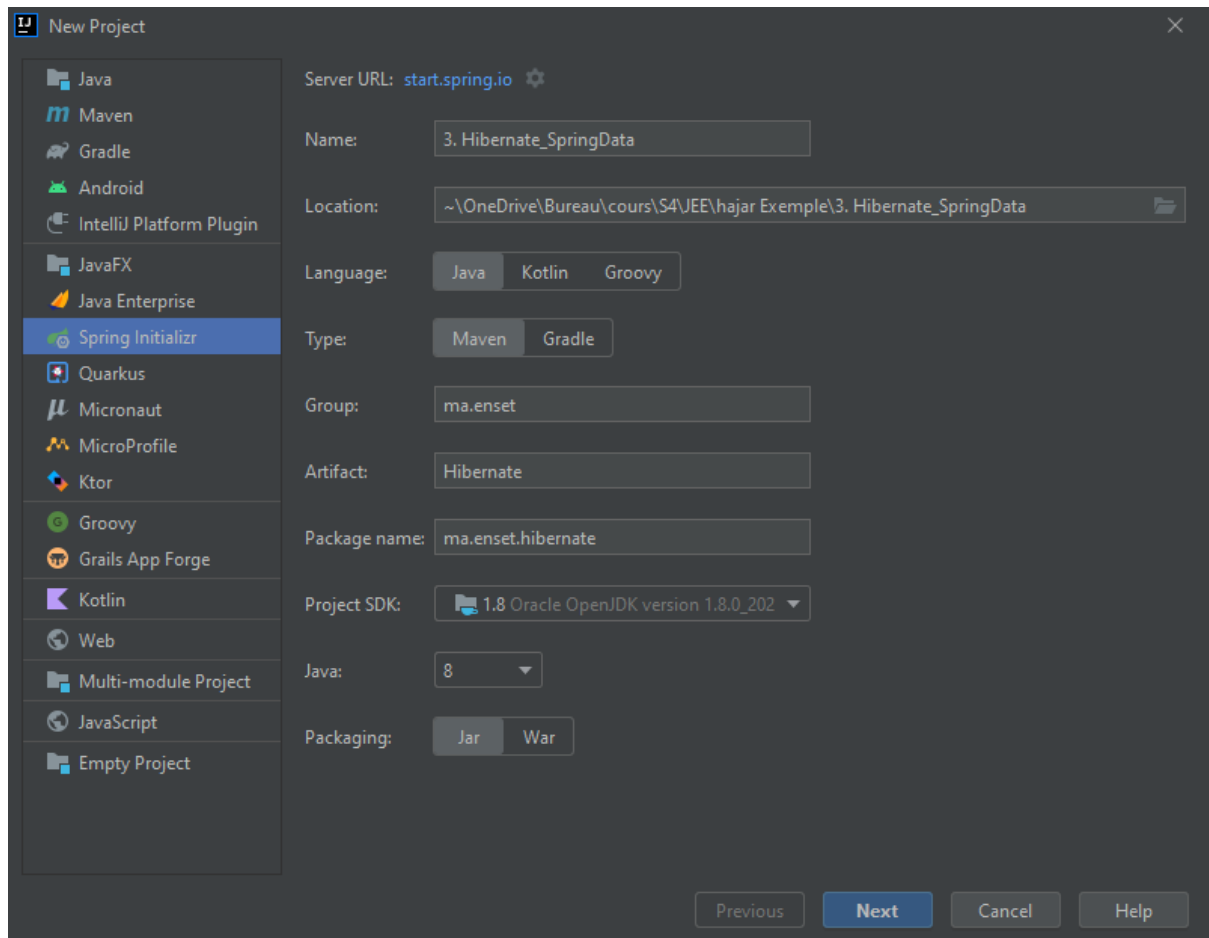


Compte rendu du TP

"JEE JPA Hibernate Spring Data"

Compte rendu rédigé par Hajar Zarguan,
Étudiante en cycle d'ingénieur: Génie des logiciels et systèmes informatiques distribuées
École Normale Supérieure d'Enseignement Technique .

1. Les captures d'écran du projet




```
HELP.md x Application.java x Patient.java x application.properties x pom.xml (Hibernate) x
1 package ma.enset.hibernate.entities;
2 import lombok.AllArgsConstructor;
3 import lombok.Data;
4 import lombok.NoArgsConstructor;
5 import javax.persistence.*;
6 import java.io.Serializable;
7 import java.util.Date;
8
9 /// Java Bean
10
11 @Entity // obligatoire pour montrer que cette classe est persistante
12 @Data // lombok va ajouter getters et setters
13 @NoArgsConstructor // lombok va ajouter un constructeur sans parametre
14 @AllArgsConstructor // lombok
15 public class Patient implements Serializable {
16     @Id // obligatoire
17     @GeneratedValue(strategy = GenerationType.IDENTITY)
18     private Long id;
19     private String nom;
20     private Date dateNaissance;
21     @Column // facultative
22     private Boolean malade;
23     private int score;
24
25 }
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help 3. Hibernate_SpringData - PatientRepository.java
3. Hibernate_SpringData src main java ma enset hibernate repositories PatientRepository
Project 3. Hibernate_SpringData C:\Users\hajar\OneDrive\B
> .idea
> .mvn
> src
> main
> java
> ma.enset.hibernate
> entities
> Patient
> repositories
> Application
> resources
1 package ma.enset.hibernate.repositories;
2 import ma.enset.hibernate.entities.Patient;
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 public interface PatientRepository extends JpaRepository <Patient, Long> {
6
7
8
9
10 }
```

```
resources application.properties
HELP.md x Application.java x Patient.java x ApplicationTests.java x PatientRepository.java x application.properties x pom.xml
1 spring.datasource.url=jdbc:h2:mem:patient-db
2 spring.h2.console.enabled=true
3 spring.datasource.driverClassName=org.h2.Driver
4 spring.datasource.username=sa
5 spring.datasource.password=
6 spring.jpa.show-sql = true
7 spring.jpa.hibernate.ddl-auto = update
8 server.port=8082
9
10
```



```
HELP.md x Application.java x Patient.java x PatientRepository.java x application.properties x n
1 package ma.enset.hibernate;
2
3 import ...
4
5
6
7 @SpringBootApplication
8 public class Application implements CommandLineRunner {
9
10     public static void main(String[] args) {
11         SpringApplication.run(Application.class, args);
12     }
13
14     @Override
15     public void run(String... args) throws Exception {}
16
17 }
18
19
```

```
1 package ma.enset.hibernate;
2
3 import ...
4
5
6
7 @SpringBootApplication
8 public class Application implements CommandLineRunner {
9     public static void main(String[] args) { SpringApplication.run(Application.class, args); }
10
11     @Autowired
12     private PatientRepository patientRepository;
13
14     @Override
15     public void run(String... args) throws Exception {
16         Date d = new Date();
17         patientRepository.save(new Patient( id: null, nom: "Hajar", d, Math.random()>0.5?true:false, score: (int)Math.random()*100 ));
18         patientRepository.save(new Patient( id: null, nom: "Hiba", d, Math.random()>0.5?true:false, score: (int)Math.random()*100 ));
19         patientRepository.save(new Patient( id: null, nom: "Hassan", d, Math.random()>0.5?true:false, score: (int)Math.random()*100 ));
20         patientRepository.save(new Patient( id: null, nom: "Ali", d, Math.random()>0.5?true:false, score: (int)Math.random()*100 ));
21         List<Patient> patients = patientRepository.findAll();
22     }
23 }
24
25
```

English Preferences Tools Help

Login

Saved Settings: Generic H2 (Embedded) v

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:~/patient-db

User Name: sa

Password:

Connect Test Connection

localhost:8082/h2-console/login.do?sessionId=0da86dedc76884c2423cdf21e1705d1f

Apps Gmail YouTube Maps PI Mind Map - Desko... Cours Mini Projet - Googl...

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:mem:patient-db

PATIENT

- ID BIGINT DEFAULT NEXT
- DATE_NAISSANCE DATE
- MALADE BOOLEAN
- NOM VARCHAR(50)
- SCORE INTEGER NOT NULL
- Indexes
- PRIMARY_KEY_F
- INFORMATION_SCHEMA
- Sequences
- Users
- H2 1.4.200 (2019-10-14)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM PATIENT

SELECT * FROM PATIENT;

ID	DATE_NAISSANCE	MALADE	NOM	SCORE
1	2022-02-28	FALSE	Hajar	0
2	2022-02-28	FALSE	Hiba	0
3	2022-02-28	TRUE	Hassan	0
4	2022-02-28	TRUE	Ali	0

(4 rows, 7 ms)

Edit

```
patientRepository.save(new Patient( Id: null, nom: "Ali", d, Math.random()>0.5?true:false, score: 9 ));

List<Patient> patients = patientRepository.findAll();
patients.forEach(p->{
    System.out.println(" ID          : " + p.getId() );
    System.out.println(" Nom          : " + p.getNom() );
    System.out.println(" Est malade : " + p.getMalade());
    System.out.println(" Score     : " + p.getScore() );
});
Patient p = patientRepository.findById(1L).orElseThrow()->new RuntimeException({});
System.out.println(p.toString());
}
```

Run: Application

Console

```
↑ Est malade : true
↓ Score      : 15
ID           : 103
Nom          : Ali
Est malade   : false
Score        : 9
Hibernate: select patient0_.id as id1_0_0_, patient0_.date_naissance as date_nai2_0_0_, patient0_.malade as malade3
Patient(Id=1, nom=Hajar0, DateNaissance=2022-02-28, malade=true, score=0)
```

repositories

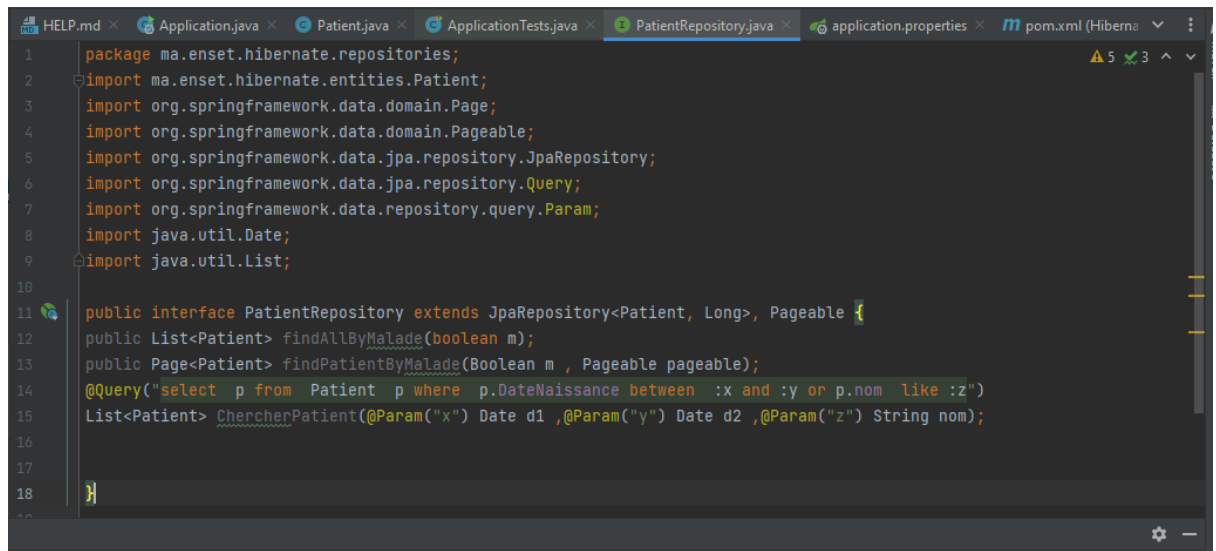
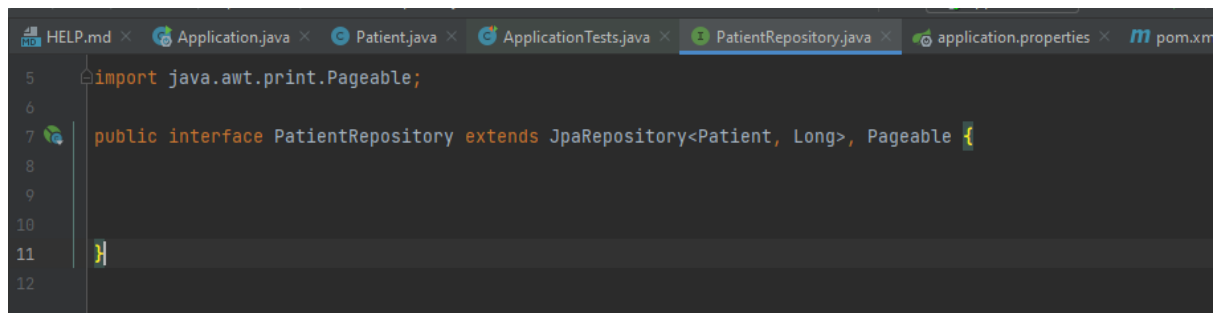
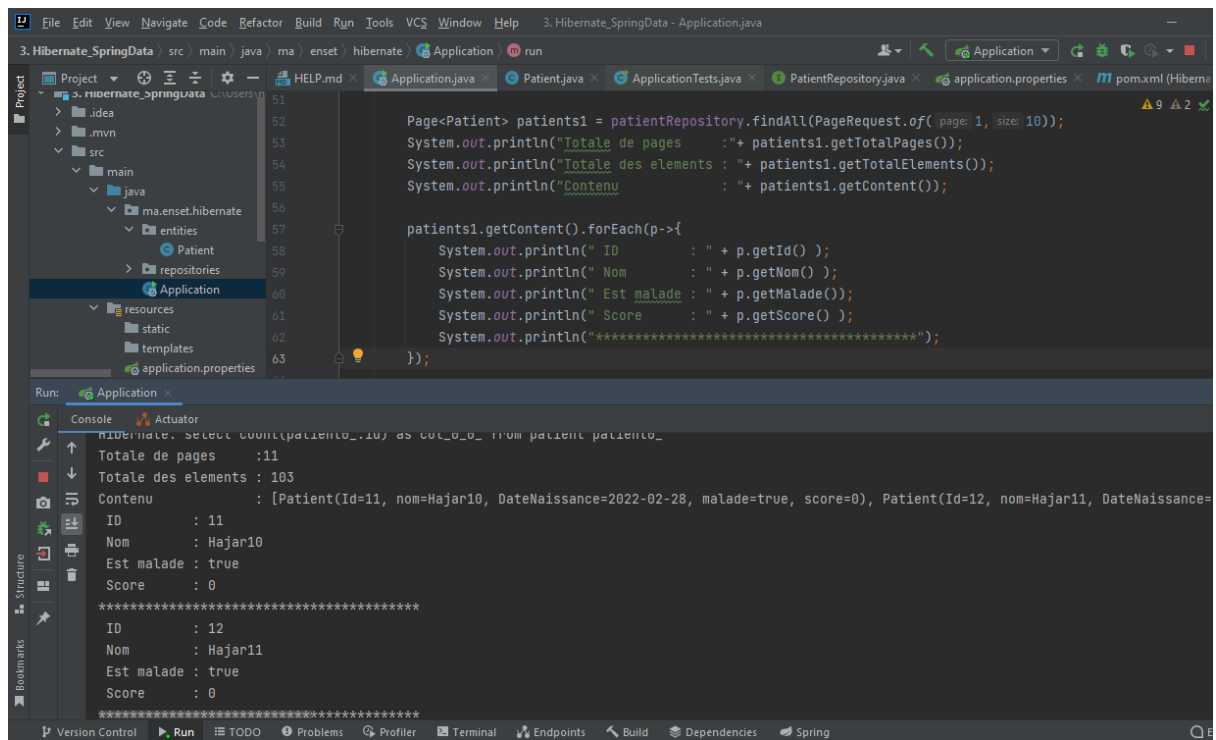
Application

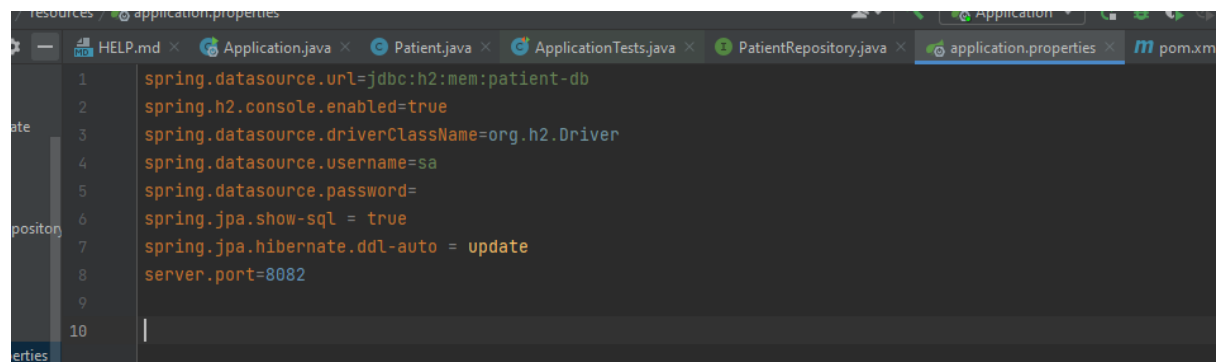
```
50
51 Page<Patient> patients1 = patientRepository.findAll(PageRequest.of( page: 1, size: 10));
52 System.out.println("Totale de pages      :"+ patients1.getTotalPages());
53 System.out.println("Totale des elements : "+ patients1.getTotalElements());
54 System.out.println("Contenu           : "+ patients1.getContent());
55
56
```

Application

Console

```
Score      : 15
ID          : 103
Nom         : Ali
Est malade  : true
Score       : 9
Hibernate: select patient0_.id as id1_0_0_, patient0_.date_naissance as date_nai2_0_0_, patient0_.malade as malade3_0_0_, patient0_.nom as nom4_0_0_, patie
Hibernate: select count(patient0_.id) as col_0_0_ from patient patient0_
Totale de pages      :11
Totale des elements : 103
Contenu              : [Patient(Id=11, nom=Hajar10, DateNaissance=2022-02-28, malade=false, score=0), Patient(Id=12, nom=Hajar11, DateNaissance=2022
```



The image shows a screenshot of an IDE window with several tabs open: HELP.md, Application.java, Patient.java, ApplicationTests.java, PatientRepository.java, application.properties (selected), and pom.xml. The application.properties file contains the following configuration:

```
1 spring.datasource.url=jdbc:h2:mem:patient-db
2 spring.h2.console.enabled=true
3 spring.datasource.driverClassName=org.h2.Driver
4 spring.datasource.username=sa
5 spring.datasource.password=
6 spring.jpa.show-sql = true
7 spring.jpa.hibernate.ddl-auto = update
8 server.port=8082
9
10
```