

# البيانات قاعدة تكوين

الوزن، خفي في نهج تطبيقي في البيانات قاعدة تعاملات تسهل قوية مكتبة هو `Spring Data JPA` تطبق في البيانات قاعدة تعاملات تسهل قوية مكتبة هو `Spring Data JPA` طريقة عمل ويتركز على `Spring Data JPA` من تعليقات أقل إنها العلائقية. البيانات قواعد مع لعمل التعليقات، على يعتمد استخدامه. كي فية حول أساسيات في داخل البيانات. قاعدة جداول في الكائنات لتزيين واضحًا، أكثر أبسط،

## 1. مشروعك إعداد.

ملف إلى الاتالية الاتباعية أضف، `pom.xml` تستخدم كنت إذا مشروعك. في `pom.xml` تضم بين عليةك العمل، لبدء بك: الخاص `pom.xml`

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jdbc</artifactId>
</dependency>
```

2- المثال، سبيل على. `Spring Data JPA` مثل بك الخاصة البيانات لقاعدة `H2` سائق إلى أيضاً سيحتاج

```
<dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <scope>runtime</scope>
</dependency>
```

المقابل: فسيكون، `Spring Data JPA` تستخدم كنت إذا

```
implementation 'org.springframework.boot:spring-boot-starter-data-jdbc'
runtimeOnly 'com.h2database:h2'
```

## البيانات قاعدة تكوين 2.

المثال، سبيل على البيانات. قاعدة اتصال بتكون قم بك، الخاص `application.yml` أو `application.properties` ملف في الذاكرة: في `2` بيانت لقاعدة

```
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.h2.console.enabled=true
```

لذلك، وفقاً المرور وكلمة المستخدم، اسم، `2` بتعديل قم، `2` مثل حقيقية بيانات لقاعدة

### 3. الـ `magical` `repository` `interface`

يتم حيـث الـ `@Service` `interface` `PersonService` `بـك`. الـ `Person` `entity` `class` `قـاعـدة` `في` `جـدول` `لـ` `Person` `مـثـيل` `بـسـيـطـة` `كـيـان` `فـئـة` `إـنـشـاء` `الـ` `أـعـمـدـة`. إـلـى الـ `الـ` `حـقـول` `يـتـم` `وـ` `الـ` `صـغـيرـة` `بـ` `الـ` `خـطـ اـفـتـرـاضـي` `بـ` `شـكـلـ` `الـ` `جـدـولـ` `اسـمـ` `إـلـى` `الـ` `فـئـةـ` `اسـمـ`

```
import org.springframework.data.annotation.Id;

public class Person {
    @Id
    private Long id;
    private String firstName;
    private String lastName;

    // Constructor          )   Spring Data JDBC)
    public Person() {}

    public Person(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    // Getters and setters
    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }
    public String getFirstName() { return firstName; }
    public void setFirstName(String firstName) { this.firstName = firstName; }
    public String getLastName() { return lastName; }
    public void setLastName(String lastName) { this.lastName = lastName; }
}
```

الـ `@Id` `مـفـاتـحـة` `يـحـددـ` `الـ` `أسـاسـيـ`. الـ `Person` `entity` `يـتـوـقـعـ` `بـ` `دونـ` `الـ` `جـدـولـ` `الـ` `Person``.`  
ذـلـكـ. تـجـاـوزـ تـمـ إـلـاـ `person` `الـ` `جـدـولـ` `تـسـمـيـةـ` `سـيـتـمـ`

### 4. `repository` `interface` `implementation`

الـ `CrudRepository` `وـ` `واجهـةـ` `تعـريـفـ` `الـ` `أسـاسـيـةـ` `الـ` `عمـلـيـاتـ` `لتـسـجـيـلـ` `CrudRepository` `تمـتـ دـوـلـةـ`

```
import org.springframework.data.repository.CrudRepository;

public interface PersonRepository extends CrudRepository<Person, Long> {
}
```

الاتشغيل. وقت فـي الـتنـفيـذـيـولـدـ [ ] تـنـفـيـذـاـ إـلـىـ تـحـتـاجـ لـاـ شـيـءـ! كلـ هـذـا

## 5. الـمـسـتـوـدـعـ اـسـتـخـدـامـ

واسـتـخـدـمـهـ [ ] أوـ خـدـمـهـ [ ] فـيـ الـمـسـتـوـدـعـ بـإـدـخـالـ قـمـ

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class PersonService {
    private final PersonRepository repository;

    @Autowired
    public PersonService(PersonRepository repository) {
        this.repository = repository;
    }

    public void savePerson() {
        Person person = new Person("John", "Doe");
        repository.save(person);
        System.out.println("Saved person with ID: " + person.getId());
    }

    public void listPeople() {
        Iterable<Person> people = repository.findAll();
        people.forEach(p -> System.out.println(p.getFirstName() + " " + p.getLastName()));
    }
}
```

## 6. تـطـبـيـقـكـ تـشـغـيـلـ

بـتـشـغـيـلـهـاـ: قـمـ وـ @SpringBootApplication معـ رـئـيـسـيـةـ فـيـةـ بـإـنـشـاءـ قـمـ، [ ] تـسـتـخـدـمـ كـنـتـ إـذـا

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Application {
    public static void main(String[] args) {
```

```
    SpringApplication.run(Application.class, args);  
}  
}
```

تلقائی! و ال مصدر بتکوین سیقوم

## مخصوصہ اسٹرالامات اختیاری: 7.

مع مستودعك إلى طرق إضافة يمكّنك مخصوصة، استعملamas إلى تحتاج لكنّت إذا @Query:

```
import org.springframework.data.jdbc.repository.query.Query;  
import org.springframework.data.repository.CrudRepository;
```

```
public interface PersonRepository extends CrudRepository<Person, Long> {  
    @Query("SELECT * FROM person WHERE first_name = :name")  
    List<Person> findByFirstName(String name);  
}
```

تذكرة يجب الـتي الرئيسيـة النقاط

```
CREATE TABLE person (
    id BIGINT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    first_name VARCHAR(255),
    last_name VARCHAR(255)
);
```

العمل سيداعلى مثال

- ت طبیق ک. ابدا
  - Person با استخدام repository.save() کائیں احفظ۔
  - repository.findAll() با استخدام اس ترجیع ادا

وسأقوم هدفك، هو ما لي فقل معين، شيء أو المخصوصة، الـ**المُعْصَمَة** أو الـ**العَلَاقَاتِ**، في الغوص تريد كنت إذا الـ**أساس**! هو هذا الـ**آن**? حتى هذا عمل كي في الـ**تفسير**. بتخصيص