

Spring Data JPA

在 Spring Boot 專案中使用存儲庫作為數據源通常涉及整合 Spring Data JPA（Java Persistence API）來管理數據庫交互。以下是如何設置的分步指南：

1. 添加依賴

首先，您需要在 `pom.xml` 文件中添加必要的依賴項，如果您使用的是 Maven。對於 Spring Data JPA，您需要以下內容：

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <scope>runtime</scope>
</dependency>
```

2. 配置數據庫

在 `application.properties` 文件中配置您的數據庫設置。例如，如果您使用的是 H2 內存數據庫：

```
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=password
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true
```

3. 創建實體

定義一個映射到數據庫表的實體類。使用 JPA 注釋來指定映射：

```
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
```

```

import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private String email;

    // Getters and setters
}

```

4. 創建存儲庫接口

創建一個擴展 JpaRepository 的存儲庫接口。該接口將為 User 實體提供 CRUD 操作：

```

import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepository extends JpaRepository<User, Long> {
    // 可以在此處定義自定義查詢方法
}

```

5. 在服務中使用存儲庫

將存儲庫注入到服務類中以執行數據庫操作：

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;

@Service
public class UserService {

    @Autowired
    private UserRepository userRepository;

    public List<User> getAllUsers() {
        return userRepository.findAll();
    }
}

```

```

public User getUserById(Long id) {
    return userRepository.findById(id).orElse(null);
}

public User saveUser(User user) {
    return userRepository.save(user);
}

public void deleteUser(Long id) {
    userRepository.deleteById(id);
}
}

```

6. 創建控制器

創建一個 REST 控制器來公開與 User 實體交互的端點：

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/users")
public class UserController {
    @Autowired
    private UserService userService;

    @GetMapping
    public List<User> getAllUsers() {
        return userService.getAllUsers();
    }

    @GetMapping("/{id}")
    public User getUserById(@PathVariable Long id) {
        return userService.getUserById(id);
    }

    @PostMapping

```

```
public User createUser(@RequestBody User user) {  
    return userService.saveUser(user);  
}  
  
@DeleteMapping("/{id}")  
public void deleteUser(@PathVariable Long id) {  
    userService.deleteUser(id);  
}  
}
```

7. 運行應用程式

運行您的 Spring Boot 應用程式。您現在應該能夠使用 REST 端點對 `User` 實體執行 CRUD 操作。

這個設置提供了一個基本結構，用於在 Spring Boot 專案中使用存儲庫作為數據源。您可以通過添加自定義查詢、處理異常和實現更複雜的業務邏輯來進一步擴展它。