

資料庫遷移與 Flyway 在 Spring Boot

這裡有一個簡潔、實用的指南，教你如何在 Spring Boot 應用中使用 Flyway 來管理你的數據庫模式：

什麼是 Flyway？

Flyway 是一個數據庫遷移工具，幫助你對數據庫模式進行版本控制。它按特定順序應用 SQL 腳本（遷移），使數據庫演變，這對於 Spring Boot 專案特別有用，因為一致性和自動化是關鍵。

第 1 步：添加 Flyway 依賴

首先，將 Flyway 添加到你的項目中。如果你使用的是 **Maven**，更新你的 `pom.xml`：

```
<dependency>
  <groupId>org.flywaydb</groupId>
  <artifactId>flyway-core</artifactId>
</dependency>
```

對於 **Gradle**，在你的 `build.gradle` 中添加以下內容：

```
implementation 'org.flywaydb:flyway-core'
```

這樣就會將 Flyway 拉入你的 Spring Boot 應用中。

第 2 步：配置數據庫連接

Flyway 需要知道要管理哪個數據庫。在 `application.properties`（或 `application.yml`）中定義你的數據庫連接。例如，使用 PostgreSQL：

```
spring.datasource.url=jdbc:postgresql://localhost:5432/mydb
spring.datasource.username=myuser
spring.datasource.password=mypassword
```

Spring Boot 會自動配置 Flyway 使用這個數據源，所以這裡不需要額外的設置。

第 3 步：創建遷移腳本

Flyway 使用版本化的 SQL 腳本來更新你的數據庫。將這些腳本放在默認目錄：`src/main/resources/db/migration`。每個腳本必須遵循這個命名模式：`V<version>__<description>.sql`。版本號確保遷移按順序運行。

例如，創建一個名為 `V1__create_users_table.sql` 的文件：

```
CREATE TABLE users (  
    id SERIAL PRIMARY KEY,  
    username VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL  
);
```

將其保存在 `src/main/resources/db/migration`。

第 4 步：運行你的應用

啟動你的 Spring Boot 應用（例如，通過 `mvn spring-boot:run` 或你的 IDE）。Flyway 會：
- 檢查數據庫的遷移歷史。
- 按順序應用任何待遷移（例如 `V1__create_users_table.sql`）。

這在啟動時自動完成，感謝 Spring Boot 的整合——無需手動干預。

第 5 步：自定義 Flyway（可選）

你可以通過在 `application.properties` 中設置屬性來調整 Flyway 的行為。這裡有一些常見的選項：

```
spring.flyway.locations=classpath:db/migration # 找到腳本的位置  
spring.flyway.baseline-on-migrate=true         # 處理現有數據庫  
spring.flyway.enabled=true                     # 打開/關閉 Flyway
```

這些選項讓你可以根據項目需求來定制 Flyway。

高級功能

Flyway 提供的不僅僅是基本的遷移：
- **基於 Java 的遷移**：為複雜邏輯編寫遷移。
- **可重複的遷移**：使用 `R__<description>.sql` 來重新運行更改的腳本（例如，視圖）。
- **撤銷遷移**：回滾更改（需要 Flyway Pro）。

查看 [Flyway 文檔](#) 以進行更深入的探索。

摘要

在 Spring Boot 中使用 Flyway：1. 將 Flyway 依賴添加到你的項目中。2. 在 `application.properties` 中設置數據庫連接。3. 在 `src/main/resources/db/migration` 中編寫版本化的 SQL 腳本。4. 啟動你的應用——Flyway 會自動應用遷移。

Flyway 通過版本控制和自動化簡化了數據庫模式管理，完美地補充了 Spring Boot 的簡潔開發體驗。