

Liquibase: 資料庫結構管理

這是一個全面的指南，教你如何在 Spring Boot 應用中使用 **Liquibase** 來有效管理資料庫結構變更：

1. 新增 Liquibase 依賴

首先，你需要在項目中包含 Liquibase 依賴。如果你使用 **Maven**，請在 `pom.xml` 中添加以下內容：

```
<dependency>
    <groupId>org.liquibase</groupId>
    <artifactId>liquibase-core</artifactId>
</dependency>
```

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

```
implementation 'org.liquibase:liquibase-core'
```

這樣會將 Liquibase 核心庫引入到你的項目中。

2. 配置 Liquibase

Liquibase 使用 **變更日誌文件** 來管理資料庫結構變更。預設情況下，Spring Boot 會在類路徑中尋找名為 `db/changelog/db.changelog-master.yaml`、`db/changelog/db.changelog-master.xml` 或 `db/changelog/db.changelog-master.yml` 的文件。你可以通過在 `application.properties`（或 `application.yml`）中添加屬性來自定義這個位置：

```
spring.liquibase.change-log=classpath:/db/changelog/db.changelog-master.xml
```

這樣告訴 Spring Boot 你的變更日誌文件的位置。

3. 創建變更日誌文件

變更日誌文件定義了你想要應用到資料庫的變更。你可以用 XML、YAML 或 SQL 格式來寫。以下是一個位於 `src/main/resources/db/changelog/db.changelog-master.xml` 的 **XML 變更日誌** 文件範例：

```

<?xml version="1.0" encoding="UTF-8"?>
<databaseChangeLog
    xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog
        http://www.liquibase.org/xml/ns/dbchangelog-dbchangelog-3.8.xsd">

    <changeSet id="1" author="your-name">
        <createTable tableName="users">
            <column name="id" type="int">
                <constraints primaryKey="true" nullable="false"/>
            </column>
            <column name="username" type="varchar(255)"/>
            <column name="email" type="varchar(255)"/>
        </createTable>
    </changeSet>
</databaseChangeLog>

```

這個範例創建了一個名為 `users` 的表，包含三個欄位：`id`、`username` 和 `email`。每個 `<changeSet>` 代表一組要應用的變更。

4. 運行你的 Spring Boot 應用

當你啟動 Spring Boot 應用時，Liquibase 會自動：- 讀取變更日誌文件。- 檢查哪些變更集已經應用（追蹤在一個名為 `DATABASECHANGELOG` 的表中）。- 將任何新的變更集應用到你的資料庫。

不需要額外的代碼——Spring Boot 的自動配置會處理這些。

5. 自定義 Liquibase (可選)

你可以使用 `application.properties` 中的屬性來調整 Liquibase 的行為。以下是一些常見的選項：

<code>spring.liquibase.enabled=true</code>	# 啟用或禁用 Liquibase
<code>spring.liquibase.drop-first=false</code>	# 在應用變更之前丟棄資料庫（請謹慎使用）
<code>spring.liquibase.contexts=dev,prod</code>	# 只在特定上下文中運行變更集

這些設置允許你將 Liquibase 適應你的環境或工作流程。

6. 利用高級功能

Liquibase 提供強大的功能來增強結構管理：
- **上下文和標籤**：控制哪些變更集在特定環境中運行（例如 dev vs. prod）。
- **預條件**：確保在應用變更集之前滿足某些條件（例如表存在）。
- **回滾**：定義如何在需要時撤銷變更集。
- **重構**：使用內建支持進行複雜的資料庫變更（例如重命名欄位）。

例如，為變更集添加上下文：

```
<changeSet id="2" author="your-name" context="dev">
    <addColumn tableName="users">
        <column name="created_at" type="timestamp"/>
    </addColumn>
</changeSet>
```

這個變更集僅在 dev 上下文中運行。

7. 使用內存資料庫進行測試

在測試中，將 Liquibase 與內存資料庫（如 H2）配對。添加 H2 依賴：

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

然後在 application.properties 中進行測試配置：

```
spring.datasource.url=jdbc:h2:mem:testdb
spring.liquibase.change-log=classpath:/db/changelog/db.changelog-master.xml
```

Liquibase 會在測試期間將你的變更應用到內存資料庫。

8. 版本控制你的變更日誌

將變更日誌文件存儲在版本控制中（例如 Git），與應用程式碼一起。這樣可以確保你的資料庫結構與代碼庫同步演進，使部署和回滾更加可預測。

摘要

在 Spring Boot 應用中使用 Liquibase：1. 新增 Liquibase 依賴。2. 配置變更日誌文件位置。3. 創建包含資料庫變更的變更日誌文件。4. 運行你的應用——Liquibase 會處理其餘部分。

Liquibase 是一個強大的工具，用於以結構化、版本控制的方式管理資料庫結構變更，並與 Spring Boot 無縫集成。