**Spring Boot: порядок загрузки в Hibernate и Flyway**

Я создал приложение Spring. Pom xml прилагается.

Он имеет конфигурацию, подобную этой (ниже), и некоторую db/migration/V1\_\_init.sql для инструмента миграции миграции Flyway.

Он имеет базу данных hsqldb в памяти и создается после запуска приложения. Он чист после создания.

Я хочу, чтобы Hibernate создавал схему на основе классов сущностей, а затем Flyway заполняла таблицы. Теперь Flyway запускает V1\_\_init.sql перед созданием таблиц и генерирует исключение. Как я могу изменить этот порядок или какое решение я могу сделать?

spring.datasource.testWhileIdle = true

spring.datasource.validationQuery = SELECT 1

spring.jpa.show-sql = true

spring.jpa.hibernate.ddl-auto = create-drop

spring.jpa.hibernate.naming-strategy = org.hibernate.cfg.ImprovedNamingStrategy

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.HSQLDialect

pom.xml:

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.3.3.RELEASE</version>

<relativePath/>

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

<version>1.3.2.RELEASE</version>

</dependency>

<dependency>

<groupId>org.hsqldb</groupId>

<artifactId>hsqldb</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>4.3.11.Final</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>4.2.5.RELEASE</version>

</dependency>

<dependency>

<groupId>org.thymeleaf</groupId>

<artifactId>thymeleaf-spring4</artifactId>

<version>2.1.4.RELEASE</version>

</dependency>

<!-- For using 'LEGACYHTML5' mode in Thymeleaf -->

<dependency>

<groupId>net.sourceforge.nekohtml</groupId>

<artifactId>nekohtml</artifactId>

<version>1.9.21</version>

</dependency>

<dependency>

<groupId>xml-apis</groupId>

<artifactId>xml-apis</artifactId>

<version>1.4.01</version>

</dependency>

<dependency>

<groupId>org.flywaydb</groupId>

<artifactId>flyway-core</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

<version>1.3.3.RELEASE</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

user3636486 08 май 2016, в 11:28

Поделиться

[Источник](https://stackoverflow.com/questions/37097876/spring-boot-hibernate-and-flyway-boot-order)

* 3

Не имеет отношения к вашему вопросу, но вы должны удалить многие версии из pom и позволить Boot настроить их для вас. И вам определенно не следует смешивать версии самой Spring Boot (у вас есть 1.3.2.RELEASE и 1.3.3.RELEASE).

Andy Wilkinson 08 май 2016, в 08:55

Spring Автоматическая настройка загрузки Flyway гарантирует, что миграция базы данных будет запущена до инициализации Hibernate. Другими словами, вы не можете полагаться на автоматическую настройку Flyway и использовать Flyway для заполнения таблиц, созданных Hinernate.

Одним из решений является полное внедрение Flyway и использование его для создания таблиц и их заполнения. Затем вы можете отключить создание таблицы Hibernate (spring.jpa.hibernate.ddl-auto=none). Этот подход более надежный, так как он позволит вашей базе данных развиваться более легко. Это то, что я бы рекомендовал вам сделать.

Другим решением является отключить автоматическую настройку Flyway (flyway.enabled=false) и настроить его самостоятельно. Затем вы можете настроить Flyway для зависимости от спящего режима, чтобы Hibernate создал таблицы до того, как Flyway попытается их заполнить.

Andy Wilkinson 08 май 2016, в 09:36

Поделиться

* 1

У вас есть пример кода вашего второго предложения?

Gabriel Oliveira 11 май 2016, в 16:54

* 0

Мы хотим, чтобы это было правдой. На самом деле Hibernate запускается первым, и это открытый вопрос.

sofend 03 янв. 2017, в 11:01

* 3

@sofend Я полагаю, что вы ошибаетесь, и вопрос также наводит на мысль. У нас также есть тесты в Boot, которые подтверждают, что миграция Flyway выполняется до Hibernate. С чего вы взяли, что Hibernate запускается первым?

Andy Wilkinson 03 янв. 2017, в 18:10

* 0

Эй, если это так, это приятные новости. У меня получилась довольно ванильная конфигурация, работающая в последней версии Spring Boot (1.4.3.RELEASE). Я использую последние аннотации тестовых рамок. Я рад поделиться некоторыми из моих настроек. Или, если у вас это работает, возможно, имеет смысл посмотреть, что у вас есть, и научиться этому. (Используете ли вы Spring Boot или просто Spring?) Проблема, с которой я сталкиваюсь, заключается в том, чтобы не запускать Flyway для заполнения данных (что и делал тот, кто создал этот поток). Я использую Flyway для переноса схемы и хочу, чтобы Hibernate проверил ее.

sofend 04 янв. 2017, в 20:20

* 0

@AndyWilkinson Flyway Flyway ensures that database migrations have run before Hibernate is initialised Удалось ли вам найти конкретный фрагмент кода, проверяющий это утверждение? Это не упоминается в их документации

krzakov 13 март 2019, в 13:09

У меня была такая же проблема.

Я хотел, чтобы моя схема была создана спящим режимом из-за независимости базы данных. Я уже столкнулся с проблемой выяснения хорошей схемы для моего приложения в моих классах jpa, мне не нравится повторять себя.

Но я хочу, чтобы какая-то инициализация данных выполнялась версией, с которой проходит Flyway.

Spring загружается миграция Flyway до спящего режима. Чтобы изменить его, я переопределил инициализатор загрузки Spring, чтобы ничего не делать. Затем я создал второй инициализатор, который запускается после спящего режима. Все, что вам нужно сделать, это добавить этот класс конфигурации:

import org.flywaydb.core.Flyway;

import org.springframework.boot.autoconfigure.flyway.FlywayMigrationInitializer;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.DependsOn;

@Configuration

public class MigrationConfiguration {

/\*\*

\* Override default flyway initializer to do nothing

\*/

@Bean

FlywayMigrationInitializer flywayInitializer(Flyway flyway) {

return new FlywayMigrationInitializer(flyway, (f) ->{} );

}

/\*\*

\* Create a second flyway initializer to run after jpa has created the schema

\*/

@Bean

@DependsOn("entityManagerFactory")

FlywayMigrationInitializer delayedFlywayInitializer(Flyway flyway) {

return new FlywayMigrationInitializer(flyway, null);

}

}

Этот код нуждается в java 8. Если у вас есть java 7 или более ранняя версия, замените (f)->{} на внутренний класс, который реализует [FlywayMigrationStrategy](http://docs.spring.io/spring-boot/docs/current/api/org/springframework/boot/autoconfigure/flyway/FlywayMigrationStrategy.html)

Конечно, вы можете сделать это в xml так же легко.

Обязательно добавьте это в свой application.properties:

flyway.baselineOnMigrate = true

user3707816 28 июнь 2017, в 15:16

Поделиться

* 2

Это прекрасно работает. Вместо @DependsOn я предпочитаю просто объявить параметр (тогда он строго типизирован), но это переопределение инициализации Flyway пути чрезвычайно полезно. Спасибо!

Paul Hilliar 25 сен. 2017, в 16:23

* 0

Хорошо, но я попробовал это, и это не может определить местонахождение моего юридического лица по его имени.

djangofan 15 окт. 2017, в 18:53

* 0

Вы добавили jpa данных Spring в ваш файл pom.xml? Проверьте, что делает автоконфигурация. Или, как сказал Пол Хиллиар, введите его по типу

user3707816 18 окт. 2017, в 18:18

* 0

это кажется устаревшим на последней версии flyway, flyway.baselineOnMigrate = true

Amare 06 нояб. 2018, в 21:29

Я решил свою задачу. Я удалил Flyway и просто добавил data.sql