

使用 MariaDB4j 给 Spring Boot 应用做集成测试

2017/10/19 1:01 - Posted By 起衣 - Tags: mariadb,spring - Category: Code

1 条评论

对于需要访问数据库的 Java 应用来说，集成测试一般都不太方便，一个项目里如果团队共享一个数据库的话，大家执行用例肯定互相干扰，而如果让每个成员本地自行搭建数据库，维护起来也很麻烦，跑用例或者构建前要去准备数据库，实在不是好的开发体验。还有如 MyBatis 推荐的那样使用内存数据库来做自动化的用例，但是可能会有潜在的不同数据库之间的差异无法被验证到，和生产环境数据库毕竟不同。

目前公司项目组使用 MySQL，个人项目也在使用 MySQL，所以最近找了下怎么做数据库相关的集成测试的资料，发现一个 MariaDB4j 的项目，把 MariaDB 封装得像一个 Java 的嵌入式数据库了，这就非常便于 Java 应用在代码里直接控制数据库的起停、创建。在这个基础上，又基于 Spring Boot 的 AutoConfiguration 机制做了一些封装，使用起来非常方便，介绍下实现。

1. 引入 Spring Boot 相关测试依赖、Junit5 Spring 扩展依赖、MariaDB4j 依赖

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-test</artifactId>
  <version>4.3.11.RELEASE</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-test</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-test-autoconfigure</artifactId>
</dependency>
<dependency>
  <groupId>com.github.sbrannen</groupId>
  <artifactId>spring-test-junit5</artifactId>
  <version>1.0.0</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>ch.vorburger.mariaDB4j</groupId>
  <artifactId>mariaDB4j</artifactId>
  <version>2.2.3</version>
  <scope>test</scope>
</dependency>
```

2. 自定义 Spring 注解

为了简化在 Spring Boot 应用下的使用，所以基于类似 Spring Boot 的 Auto Configure 机制做了封装，达到只需要在测试类上添加一个注解，就可以自行启动一个

MariaDB 数据库，并创建指定的 database 的目的。

首先创建一个 MariaDB4jFactoryBean，继承了 MariaDB4j 项目里的 MariaDB4jSpringService，这个 Bean 根据配置信息在启动数据库后创建指定的 database，然后实现一个注解 AutoConfigureMariaDB，处理此注解的类为 TestMariaDBAutoConfiguration，TestMariaDBAutoConfiguration 是一个 Spring 的 Java Configuration 类，在这里暴露 MariaDB4jFactoryBean。最后在 META-INF/spring.factories 里把注解和类实现给添加进去，这样 Spring Boot 初始化的时候，就会根据注解找到对应的 Configuration 类，并自动暴露添加的 Spring Bean 了，并且 Auto Configuration 还可以指定配置的顺序，这里创建数据库就指定在 DataSourceAutoConfiguration 之前了。具体的代码实现可以参考：<https://github.com/momoment/pink/tree/master/src/test/java/com/momo/pink/test/autoconfigure>

3. 写一个集成测试用例

```
@ExtendWith(SpringExtension.class)
@SpringBootTest(classes = PinkApp.class, properties = {
    "spring.test.maria.database=pink",
    "spring.datasource.username=root",
    "spring.datasource.password=",
}, webEnvironment = SpringBootTest.WebEnvironment.RANDOM_PORT)
@AutoConfigureMariaDB
public class UserTest {

    @Autowired
    private TestRestTemplate restTemplate;

    @Test
    public void testCreateUser() {
        User user = this.restTemplate.postForObject(
            "/api/v1.0/users", new User()
                .setEmail("bphanzhu@gmail.com")
                .setName("qiyi"), User.class);
        assertNotNull(user.getId());
        User queryUser = this.restTemplate.getForObject(
            "/api/v1.0/users/" + user.getName(), User.class);
        assertEquals(user.getId(), queryUser.getId());
        assertEquals(user.getName(), queryUser.getName());
        assertEquals(user.getEmail(), queryUser.getEmail());
    }
}
```

使用了 Junit5 的 ExtendWith 来启动 SpringBoot 应用，SpringBootTest 注解指定了应用入口类，以及测试环境下的一些自定义配置，同时要求真实地启动 tomcat 环境，最后一个 @AutoConfigureMariaDB 注解就直接把数据库给创建起来了，配合 Flyway 的集成可以自动完成数据库表的创建。

本文链接: <http://isouth.org/archives/361.html>，转载请注明出处，此外还可以[订阅我](#)。

相关日志 Relate Posts

- [SpringBoot应用配置项加密](#)

- SpringWeb基于请求参数值分发HTTP请求
- SpringBoot 集成 Flyway 自动创建数据库表

收藏与分享 : [Twitter](#) | [Facebook](#) | [微博](#) | [人人](#) | [Google+](#) | [PDF](#)

发表留言 (Ctrl+Enter提交)

昵称 (必填)

EMail (不公开, 必填)

网站 (选填)



发表

themed by [white](#) 1.18 hosted by [衡天主机](#) copyright © 2017 [isouth.org](#) 19 queries in 0.392 seconds.

[回到顶端](#) ↑