# 添加依赖包

<dependency>

<groupId>org.dbunit</groupId>

<artifactId>dbunit</artifactId>

<version>2.5.2</version>

<type>jar</type>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.github.springtestdbunit</groupId>

<artifactId>spring-test-dbunit</artifactId>

<version>1.1.0</version>

<scope>test</scope>

</dependency>

# 编写xml文件用于插入数据

<TestPerson.xml>

### 格式如下：

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<-- 固定的!-->

<dataset>

<-- test 为表名，user\_name为列名 c为插入值-->

<testuser\_name=*"c"*/>

<testuser\_name=*"d"*/>

</dataset>

# 在mysql中添加测试数据库

C:\Users\KR0053\AppData\Local\Temp\1520825595(1).png

# 编写Spring-boot的配置类

package com.stylefeng.guns.config;

import javax.sql.DataSource;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import com.github.springtestdbunit.bean.DatabaseConfigBean;

import com.github.springtestdbunit.bean.DatabaseDataSourceConnectionFactoryBean;

@Configuration

public class DBUnitConfiguration {

@Bean

public DatabaseConfigBean dbUnitDatabaseConfig() {

DatabaseConfigBean bean = new DatabaseConfigBean();

return bean;

}

@Bean

public DatabaseDataSourceConnectionFactoryBean dbUnitDatabaseConnection(DatabaseConfigBean dbUnitDatabaseConfig,

DataSource dataSource) {

DatabaseDataSourceConnectionFactoryBean bean = new DatabaseDataSourceConnectionFactoryBean(dataSource);

bean.setDatabaseConfig(dbUnitDatabaseConfig);

return bean;

}

}

# 编写dao的集成测试类

package test.com.stylefeng.guns.modular.bom.service;

import static org.hamcrest.MatcherAssert.assertThat;

import static org.hamcrest.Matchers.hasSize;

import static org.hamcrest.Matchers.notNullValue;

import java.util.List;

import javax.annotation.Resource;

import javax.sql.DataSource;

import org.dbunit.DBTestCase;

import org.dbunit.database.DatabaseConnection;

import org.dbunit.database.IDatabaseConnection;

import org.dbunit.dataset.IDataSet;

import org.junit.Before;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.boot.test.mock.mockito.MockitoTestExecutionListener;

import org.springframework.test.context.TestExecutionListeners;

import org.springframework.test.context.junit4.AbstractTransactionalJUnit4SpringContextTests;

import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import org.springframework.test.context.support.DependencyInjectionTestExecutionListener;

import org.springframework.test.context.support.DirtiesContextTestExecutionListener;

import org.springframework.test.context.transaction.TransactionConfiguration;

import org.springframework.test.context.transaction.TransactionalTestExecutionListener;

import org.springframework.transaction.annotation.Transactional;

import com.github.springtestdbunit.DbUnitTestExecutionListener;

import com.github.springtestdbunit.annotation.DatabaseSetup;

import com.stylefeng.guns.TestApplication;

import com.stylefeng.guns.modular.bom.service.IBomService;

import com.stylefeng.guns.modular.system.model.Bom;

@SuppressWarnings("deprecation")

@RunWith(SpringJUnit4ClassRunner.class)

//测试的启动类

@SpringBootTest(classes = { TestApplication.class })

//指定 dbunit的listener：

@TestExecutionListeners({

DependencyInjectionTestExecutionListener.class,

DirtiesContextTestExecutionListener.class,

TransactionalTestExecutionListener.class,

MockitoTestExecutionListener.class,

DbUnitTestExecutionListener.class

})

@Transactional

@TransactionConfiguration(transactionManager = "transactionManager", defaultRollback = true)

public class TestBomServiceImpl {

@Autowired

IBomService bomService;

@Test

@DatabaseSetup({"classpath:data/bom.xml"})

public final void testList(){

List<Bom> boms = bomService.selectList(null);

for(Bom bom : boms){

System.out.println(bom);

}

assertThat(boms, hasSize(2));

}

@Test

public final void testInsert(){

Bom bom = new Bom();

bom.setBatchBomFlag(false);

bom.setBatchName("test");

bom.setBomName("testBomName");

bomService.insert(bom);

assertThat(bom.getId() , notNullValue());

}

}