第十章习题答案

1.填空题

（1）原子性、一致性、隔离性、持久性

（2）TransactionDefinition

（3）TransactionStatus

（4）PlatformTransactionManager

（5）TransactionTemplate

2.选择题

（1）B

（2）C

（3）A

（4）C

（5）A

3.思考题

（1）

答：Spring主要通过三个接口实现事务抽象，这三个接口分别是TransactionDefinition、TransactionStatus、PlatformTransactionManager，它们都位于org.springframework.transaction包中，其中，TransactionDefinition用于定义事务的属性，TransactionStatus用于界定事务的状态，PlatformTransactionManager根据属性管理事务。

（2）

答：在Spring中，除了基于XML配置声明式事务，还可以通过注解配置声明式事务。@Transactional是Spring提供的配置声明式事务的主要注解，它可以实现和XML文件中<tx:advice>元素相同的功能。

4.编程题

Ⅰ.创建pojo类Stu.java

1. package com.qfedu.pojo;
2. public class Stu {
3. private Integer sid;
4. private String sname;
5. private String age;
6. private String course;
7. public Stu() {
8. super();
9. // TODO Auto-generated constructor stub
10. }
11. public Stu(Integer sid, String sname, String age, String course) {
12. super();
13. this.sid = sid;
14. this.sname = sname;
15. this.age = age;
16. this.course = course;
17. }
18. public Integer getSid() {
19. return sid;
20. }
21. public void setSid(Integer sid) {
22. this.sid = sid;
23. }
24. public String getSname() {
25. return sname;
26. }
27. public void setSname(String sname) {
28. this.sname = sname;
29. }
30. public String getAge() {
31. return age;
32. }
33. public void setAge(String age) {
34. this.age = age;
35. }
36. public String getCourse() {
37. return course;
38. }
39. public void setCourse(String course) {
40. this.course = course;
41. }
42. @Override
43. public String toString() {
44. return "Stu [sid=" + sid + ", sname=" + sname + ", age=" + age + ", course=" + course + "]";
45. }
46. }

Ⅱ.创建dao

1. package com.qfedu.dao;
2. import java.util.List;
3. import com.qfedu.pojo.Stu;
4. public interface StuDao {
5. //删除
6. void deleteByName(String name);
7. //增加
8. void addStu(Stu stu);
9. //查询
10. List<Stu> selectAll();
11. }

Ⅲ.创建dao.impl

1. <?xml version="1.0" encoding="UTF-8"?>
2. <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "../mybatis-3-mapper.dtd" >
3. <mapper namespace="com.qfedu.dao.StuDao">
4. <delete id="deleteByName" >
5. delete from stu where sname=#{sname}
6. </delete>
7. <insert id="addStu" useGeneratedKeys="true" keyProperty="sid">
8. insert into stu(sid,sname,age,course)
9. values(#{sid},#{sname},#{age},#{course})
10. </insert>
11. <select id="selectAll" resultType="com.qfedu.pojo.Stu">
12. select \* from stu
13. </select>
14. </mapper>

Ⅳ.创建service

1. package com.qfedu.service;
2. import com.qfedu.pojo.Stu;
3. public interface StuService {
4. public String saveOther(String delStuName,Stu stu);
5. }

Ⅴservice.impl

1. package com.qfedu.service.impl;
2. import javax.annotation.Resource;
3. import org.springframework.stereotype.Service;
4. import org.springframework.transaction.annotation.Transactional;
5. import com.qfedu.dao.StuDao;
6. import com.qfedu.pojo.Stu;
7. import com.qfedu.service.StuService;
8. @Service
9. public class StuServiceImpl implements StuService {
10. @Resource
11. private StuDao stuDao;
12. @Override
13. @Transactional
14. public String saveOther(String delStuName, Stu stu) {
15. stuDao.deleteByName(delStuName);
16. if(true) {
17. throw new RuntimeException();
18. }
19. stuDao.addStu(stu);
20. return "success";
21. }
22. }.

Ⅵ.application.xml开启事务

1. <?xml version="1.0" encoding="UTF-8"?>
2. <beans xmlns="http://www.springframework.org/schema/beans"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4. xmlns:context="http://www.springframework.org/schema/context"
5. xmlns:aop="http://www.springframework.org/schema/aop"
6. xmlns:tx="http://www.springframework.org/schema/tx"
7. xsi:schemaLocation="http://www.springframework.org/schema/beans
8. http://www.springframework.org/schema/beans/spring-beans.xsd
9. http://www.springframework.org/schema/context
10. http://www.springframework.org/schema/context/spring-context.xsd
11. http://www.springframework.org/schema/aop
12. http://www.springframework.org/schema/aop/spring-aop.xsd
13. http://www.springframework.org/schema/tx
14. http://www.springframework.org/schema/tx/spring-tx.xsd">
15. <context:component-scan base-package="com.qfedu.\*" />
16. <!-- 引入外部 properties 文件 -->
17. <context:property-placeholder location="classpath:jdbc.properties"/>
18. <!-- 注册数据源 -->
19. <bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">
20. <property name="driverClass" value="com.mysql.jdbc.Driver"/>
21. <property name="jdbcUrl" value="jdbc:mysql://localhost:3306/chapter10?characterEncoding=UTF-8"/>
22. <property name="user" value="root"/>
23. <property name="password" value="root"/>
24. </bean>
25. <!--3 会话工厂bean sqlSessionFactoryBean -->
26. <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
27. <!-- 数据源 -->
28. <property name="dataSource" ref="dataSource"></property>
29. <!-- <property name="configLocation" value="classpath:mybatis-config.xml"></property> -->
30. <!-- sql映射文件路径 -->
31. <property name="mapperLocations" value="classpath\*:com/qfedu/dao/impl/\*Mapper.xml"></property>
32. </bean>
33. <!-- 自动扫描对象关系映射 -->
34. <!--创建dao-->
35. <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
36. <!--指定会话工厂，如果当前上下文中只定义了一个则该属性可省去 -->
37. <property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"></property>
38. <!-- 指定要自动扫描接口的基础包，实现接口 -->
39. <property name="basePackage" value="com.qfedu.dao"></property>
40. </bean>
42. <bean id="txManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
43. <property name="dataSource" ref="dataSource"/>
44. </bean>
45. <tx:annotation-driven transaction-manager="txManager"/>
46. </beans>

Ⅶ.test测试类

1. package com.qfedu.test;
2. import javax.annotation.Resource;
3. import org.junit.Test;
4. import org.junit.runner.RunWith;
5. import org.springframework.test.context.ContextConfiguration;
6. import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
7. import com.qfedu.pojo.Stu;
8. import com.qfedu.service.StuService;
9. @RunWith(SpringJUnit4ClassRunner.class)
10. @ContextConfiguration(locations = {"application.xml"})
11. public class Test1 {
12. @Resource
13. private StuService stuService;
14. @Test
15. public void test() {
16. stuService.saveOther("zhangsan",new Stu(null,"LiSi","20","java"));
17. }
18. @Test
19. public void test1() {
20. stuService.saveOther("LiSi",new Stu(null,"zhangsan","20","java"));
21. }
22. }