Linux认证:SpringJDBC事务传播特性Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/559/2021_2022_Linux_E8_AE_A4_E8_AF_c104_559606.htm Spring 和 JDBC 整合开发：（2）通过JDBC模板类 处理 异常 处理 事务的传播特性 处理事务的隔离性 @Transactional(noRollbackFor=RuntimeException.class) public void save(User user) throws Exception{ // TODO Auto-generated method stub this.template.0update("insert into user values(?,?)", new Object[]{new Integer(user.getId()),user.getName()}, new int[]{java.sql.Types.INTEGER,java.sql.Types.VARCHAR}). System.out.println("插入成功...."). throw new RuntimeException(). } 我们制定了参数：noRollbackFor 所以 即使 遇到了运行期异常（nocheck异常）的时候 仍然不会回滚 Spring默认的是 只是 运行期异常才会回滚但是 我们这里可以认为的指定哪些类 需要回滚事务 @Transactional(rollbackFor=Exception.class) public void save(User user) throws Exception{ // TODO Auto-generated method stub this.template.0update("insert into user values(?,?)", new Object[]{new Integer(user.getId()),user.getName()}, new int[]{java.sql.Types.INTEGER,java.sql.Types.VARCHAR}). System.out.println("插入成功...."). throw new Exception(). } 此时我们指定Exception异常要回滚 所以 执行结果是事务回滚了 下面是 事务传播特性的 第一个例子：Required nerver给出代码 ： @Transactional(propagation=Propagation.REQUIRED) public void save(User user) { // TODO Auto-generated method stub

this.template.0update("insert into user values(?,?)", new Object[]{new Integer(user.getId()),user.getName()}, new int[]{java.sql.Types.INTEGER,java.sql.Types.VARCHAR}). System.out.println("插入成功...."). userdao_1.save(null). } 代码二：@Transactional(propagation=Propagation.NEVER) public void save(User user){ // TODO Auto-generated method stub this.getTemplate().0update("insert into user values(21,fasd)"). System.out.println("插入成功....?????????????///"). } 代码一 需要事务 当执行到dao层是 检测到没有事务 所以 就 给分配了事务：但是 嵌套了一个方法，代码二中的方法不需要事务PROPAGATION.NEVER但是 它运行在了事务环境中 所以 抛出异常 因为 代码二：运行在代码一的事务环境中，出现了异常（此异常是RuntimeExcpetion的一个子类） 所以 全部回滚 一条数据 都没有插入进； 强制：代码一：@Transactional(propagation=Propagation.NEVER) public void save(User user) { // TODO Auto-generated method stub this.template.0update("insert into user values(?,?)", new Object[]{new Integer(user.getId()),user.getName()}, new int[]{java.sql.Types.INTEGER,java.sql.Types.VARCHAR}). System.out.println("插入成功...."). userdao_1.save(null). } 代码二：@Transactional(propagation=Propagation.MANDATORY) public void save(User user){ // TODO Auto-generated method stub this.getTemplate().0update("insert into user values(21,fasd)"). System.out.println("插入成功....?????????????///"). } 运行结果抛出异常：但是 代码一不需要事务 运行完SQL 就对数据库 修改了，但是代码二需要事务 但是运行在没有事务的环境下所以

跑出了异常 支持： @Test public void save() throws Exception { BeanFactory bf = new ClassPathXmlApplicationContext("applicationContext.xml"). UserDAO dao =(UserDAO) bf.getBean("userdao"). User user = new User(). user.setId(11). user.setName("刘强"). dao.save(user). } 代码一： @Transactional(propagation=Propagation.NEVER) public void save(User user) { // TODO Auto-generated method stub this.template.0update("insert into user values(?,?)", new Object[]{new Integer(user.getId()),user.getName()}, new int[]{java.sql.Types.INTEGER,java.sql.Types.VARCHAR}). System.out.println("插入成功...."). userdao_1.save(null). } 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com