DeclarativeTransactionDemo

```
-----tarena
                    |----spring-config.xml
          <!-- 加入Spring2Core -->
          <!-- 加入Spring2AOP -->
          <!-- 加入Spring2PersistenceCore -->
          <!-- 加入Spring2PersistenceJDBC -->
          <!-- 加入MySQL -->
步骤一,
public class User {
          private int id;
          private String name;
          public User() {
          public int getId() {
                    return id;
          }
          public void setId(int id) {
                    this.id = id;
          }
          public String getName() {
                               第1页
```

```
实验三.txt
                    return name;
          }
          public void setName(String name) {
                    this.name = name;
          }
}
步骤二,
public interface IUserDao {
          public void find(int id);
          public void insert(User user);
          public void update(User user);
          public void delete(User user);
}
步骤三,
public class UserDaoImpl implements IUserDao {
          // 关联的属性
          private JdbcTemplate jt;
          // 注入方法
          public void setDataSource(DataSource dataSource) {
                    jt = new JdbcTemplate(dataSource);
                    if (jt != null) {
                              System.out.println("jt模板创建成功");
                    }
          }
          public void delete(User user) {
                    System.out.println("正在删除用户。。。");
          }
          @Transactional(readOnly = true)
          public void find(int id) {
                              第2页
```

```
实验三.txt
                    System.out.println("正在查询用户。。。");
                   String sql = "SELECT * FROM user WHERE
user_id = ?";
                    Object[] args = \{ id \};
                   List<User> users = jt.queryForList(sql, args);
                    // 返回值为一个java.util.LinkedHashMap类
                   Iterator it = users.iterator();
                   while (it.hasNext()) {
                             Map userMap = (Map) it.next();
System.out.print(userMap.get("user_id") + "\t");
System.out.print(userMap.get("user_name") + "\t");
System.out.println(userMap.get("user_gender"));
          }
          @Transactional(propagation = Propagation.REQUIRED)
          public void insert(User user) {
                   System.out.println("正在增加用户。。。");
                   String sql = "INSERT INTO
user(user_name,user_gender) VALUES(?,?)";
                   Object[] args = { user.getUser_name(),
user.getUser_gender() };
                   int i = jt.update(sql, args);
                    if (i > 0)
                             System.out.println("插入成功");
                              // 定会抛出异常,事务回滚
                              //Integer.parseInt("abc123");
                   }
          }
          public void update(User user) {
                   System.out.println("正在修改用户。。。");
         }
}
```

实验三.txt

```
步骤四,
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"</pre>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="http://www.springframework.org/schema/bea
ns
http://www.springframework.org/schema/beans/spring-beans-2.0.x
sd
         http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-2.0.xsd">
         <!-- 修改XML头信息 -->
         <!-- 带连接池功能的
org.apache.commons.dbcp.BasicDataSource -->
         <!-- 数据源 -->
         <bean id="DriverManagerDataSource"</pre>
class="org.springframework.jdbc.datasource.DriverManagerDataSour
ce">
                  property name="driverClassName">
<value>com.mysql.jdbc.Driver</value>
                  coperty name="url">
<value>jdbc:mysql://localhost:3306/test</value>
                  </property>
                  property name="username">
                           <value>root</value>
                  </property>
                  property name="password">
                           <value>admin</value>
                  </bean>
```

实验三.txt

<bean id="DataSourceTransactionManager"</pre>

<!-- 数据源事务管理器 -->

class="org.springframework.jdbc.datasource.DataSourceTransaction Manager"> property name="dataSource"> <ref bean="DriverManagerDataSource"</pre> /> </bean> <!-- 持久层 --> <bean id="UserDaoImpl" class="tarena.UserDaoImpl"> property name="dataSource"> <ref bean="DriverManagerDataSource"</pre> /> </property> </bean> <!-- Spring2声明事务管理 --> <tx:annotation-driven transaction-manager="DataSourceTransactionManager" /> </beans> 步骤五, public class Test { public static void main(String[] args) { ApplicationContext context = new ClassPathXmlApplicationContext("tarena/spring-config.xml"); IUserDao userDao = (IUserDao)context.getBean("UserDaoImpl");

第5页

实验三.txt

```
//userDao.find(5);

User user = new User();
    user.setUser_name("zhaojun");
    user.setUser_gender("m");

userDao.insert(user);
}
```