

一，声明式事务管理（注解）

1，配置事务管理器

<!-- 第一步 配置事务管理器 -->

```
<bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    <property name="dataSource" ref="dataSource"> </property>
</bean>
```

2，配置事务注解

<!-- 第二步 开启事务注解 -->

```
<tx:annotation-driven transaction-manager="transactionManager"/>
```

3，在要使用事务的方法所在的类上面添加注解

```
@Transactional
public class OrderService {
    private OrderDao orderDao;
```

二，具体的代码

1，orderDao实体类

```
package cn.java.ZhuanZhang.Dao;
import org.springframework.jdbc.core.JdbcTemplate;
public class OrderDao {
    //注入jdbcTemplate
    private JdbcTemplate jdbcTemplate;
    public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {
        this.jdbcTemplate = jdbcTemplate;
    }
    /**
     * 做对数据库操作的方法，不写业务操作
     */
    //少钱的方法
    public void lessMoney(){
        String string = "update user set salary = salary-? where username=?";
        jdbcTemplate.update(string,100,"anqili");
    }
    //多钱的方法
    public void moreMoney(){
        String string = "update user set salary = salary+? where username=?";
        jdbcTemplate.update(string,100,"linmingjun");
    }
}
```

2，orderService实体类

```
package cn.java.ZhuanZhang.Service;
import org.springframework.transaction.annotation.Transactional;
```

```

import cn.java.ZhuanZhang.Dao.OrderDao;
@Transactional
public class OrderService {
    private OrderDao orderDao;
    public void setOrderDao(OrderDao orderDao) {
        this.orderDao = orderDao;
    }
    //调用dao实体类中的方法
    //业务逻辑层
    public void accountMoney(){
        //anqili少100元
        orderDao.lessMoney();
        int i = 100/0;
        //linmingjun多100元
        orderDao.moreMoney();
    }
}

```

3, bean.xml 配置文件

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:jdbc="http://www.springframework.org/schema/jdbc"
    xmlns:jee="http://www.springframework.org/schema/jee"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:util="http://www.springframework.org/schema/util"
    xmlns:task="http://www.springframework.org/schema/task"
    xmlns:aop="http://www.springframework.org/schema/aop"

    xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        https://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/tx
        http://www.springframework.org/schema/tx/spring-tx.xsd
        http://www.springframework.org/schema/aop
        https://www.springframework.org/schema/aop/spring-aop.xsd
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context-4.0.xsd
        http://www.springframework.org/schema/jdbc
        http://www.springframework.org/schema/jdbc/spring-jdbc-4.0.xsd
        http://www.springframework.org/schema/jee
        http://www.springframework.org/schema/jee/spring-jee-4.0.xsd
        http://www.springframework.org/schema/tx
        http://www.springframework.org/schema/tx/spring-tx-4.0.xsd

```

```

http://www.springframework.org/schema/util
http://www.springframework.org/schema/util/spring-util-4.0.xsd
http://www.springframework.org/schema/task
http://www.springframework.org/schema/task/spring-task-4.0.xsd
    default-lazy-init="false">
    <!-- 配置c3p0的连接池 -->
    <bean id="dataSource"
class="com.mchange.v2.c3p0.ComboPooledDataSource">
        <!-- 注入属性值 -->
        <property name="driverClass" value="com.mysql.cj.jdbc.Driver">
</property>
        <property name="jdbcUrl"
            value="jdbc:mysql://localhost:3306/hb1?serverTimezone=UTC">
</property>
        <property name="user" value="root"></property>
        <property name="password" value="AQL271422"></property>
    </bean>
    <!-- 第一步 配置事务管理器 -->
    <bean id="transactionManager"

class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
        <property name="dataSource" ref="dataSource"></property>
    </bean>
    <!-- 第二步 开启事务注解 -->
    <tx:annotation-driven transaction-manager="transactionManager" />

    <!-- create object -->
    <bean id="orderDao" class="cn.java.ZhuanZhang.Dao.OrderDao">
        <!-- 注入jdbcTemplate -->
        <property name="jdbcTemplate" ref="jdbcTemplate"></property>
    </bean>
    <bean id="orderService" class="cn.java.ZhuanZhang.Service.OrderService">
        <!-- 注入属性 -->
        <property name="orderDao" ref="orderDao"></property>
    </bean>
    <!-- 注入模板 -->
    <bean id="jdbcTemplate"
class="org.springframework.jdbc.core.JdbcTemplate">
        <!-- 把dataSource传递到模板对象里面 -->
        <property name="dataSource" ref="dataSource"></property>
    </bean>
</beans>

```

4, 测试代码

```

package cn.java.testSet;
import org.junit.Test;
import org.springframework.context.ApplicationContext;

```

```
import org.springframework.context.support.ClassPathXmlApplicationContext;
import cn.java.ZhuanZhang.Service.OrderService;
public class TestZhuanZhang {
    @Test
    public void testTemplate(){
        ApplicationContext context =
            new ClassPathXmlApplicationContext("bean2.xml");
        OrderService book = (OrderService)context.getBean("orderService");
        book.accountMoney();
    }
}
```