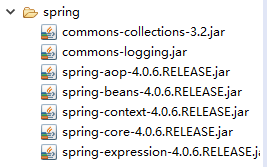
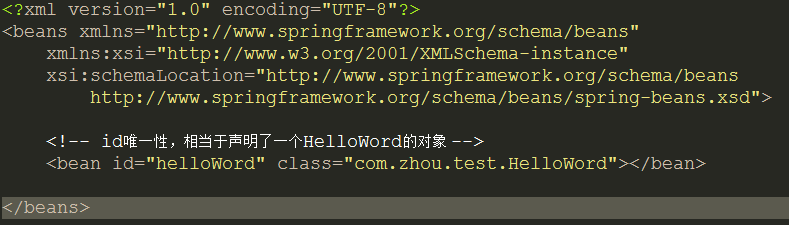
**使用Spring前准备：**

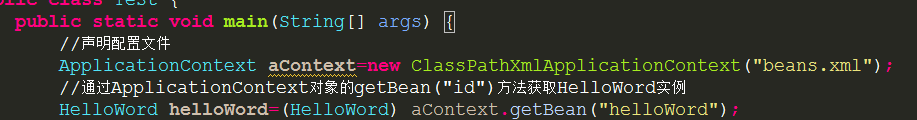
把核心包拉进项目lib文件夹下，创建beans.xml文件



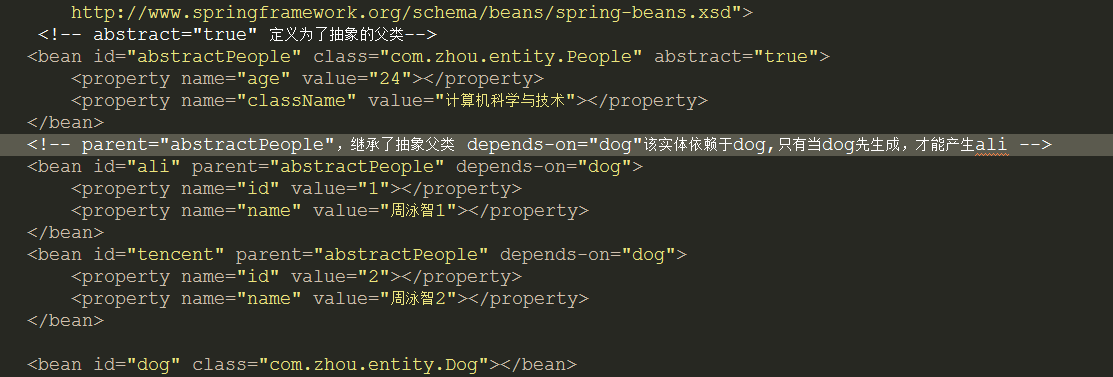
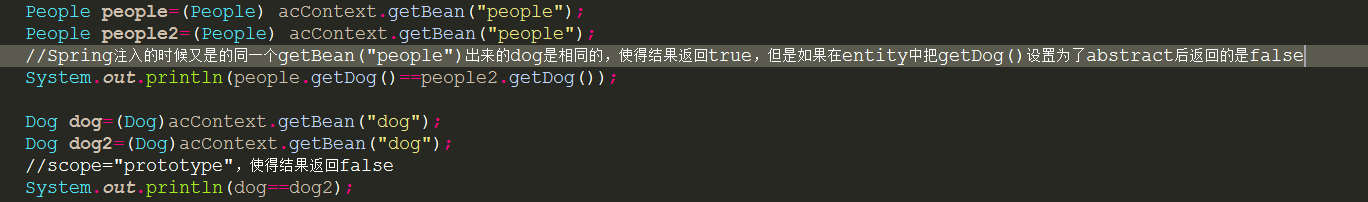
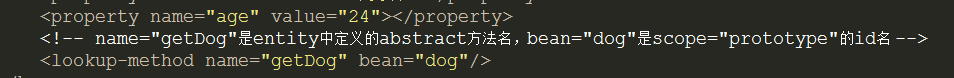
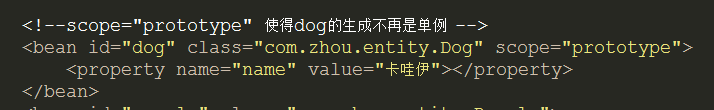
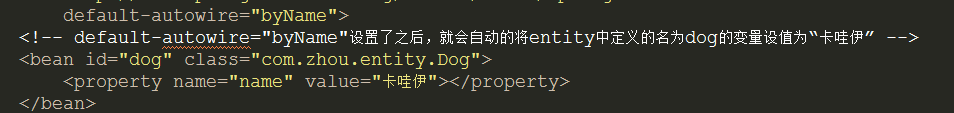
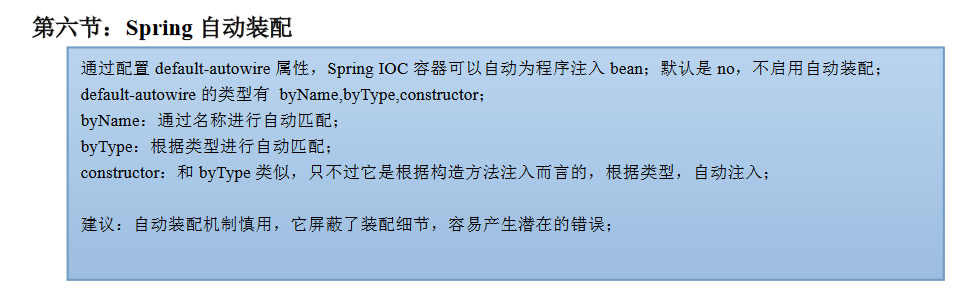
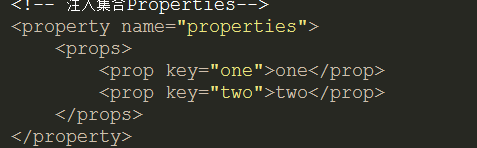
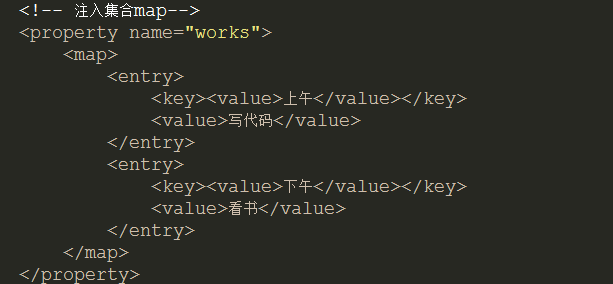
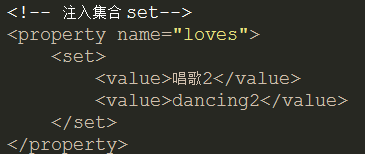
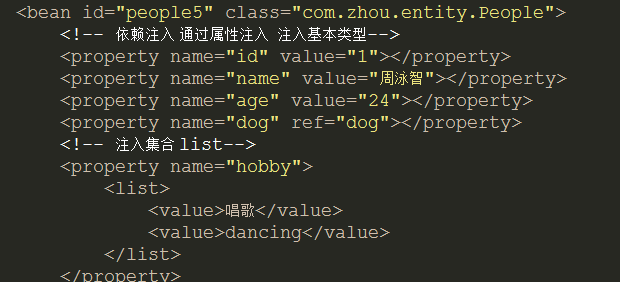
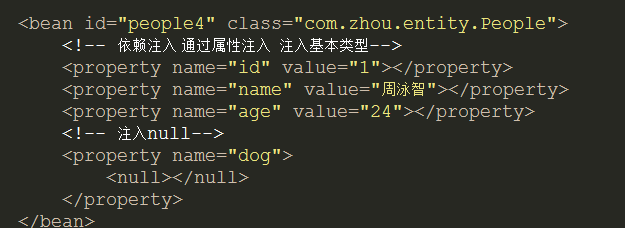
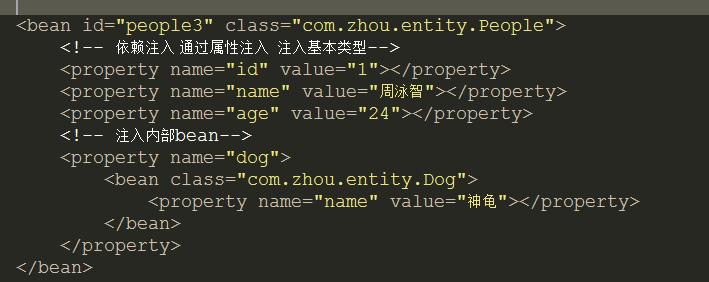
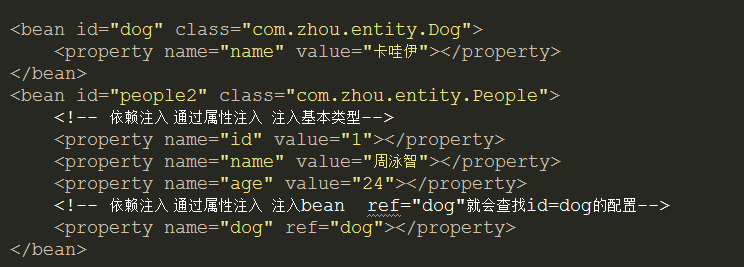
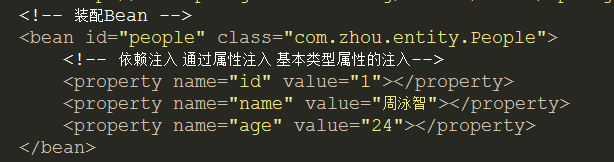
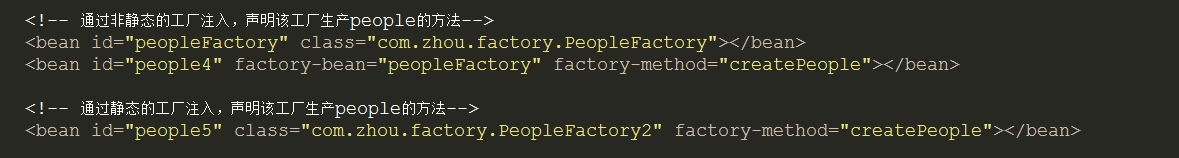
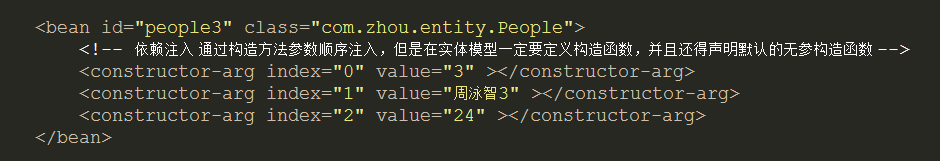
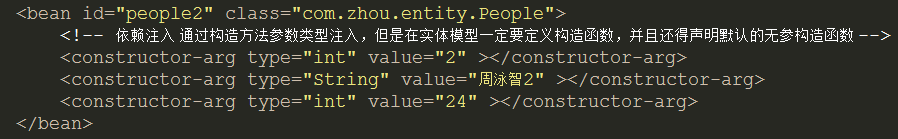
Spring之Helloword

配置beans.xml

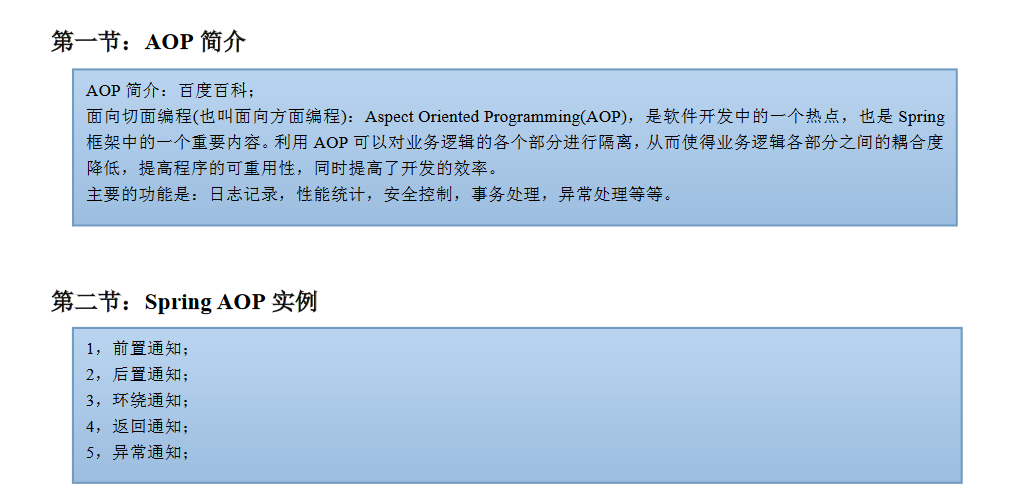




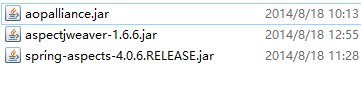
第二节Spring之ioc详解



第三节AOP详解



使用AOP要先引入AOP必须的三个包



<?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:aop="http://www.springframework.org/schema/aop"

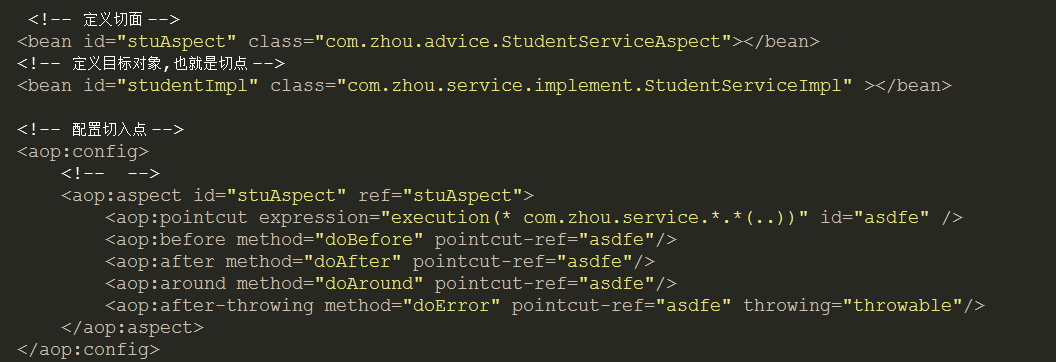
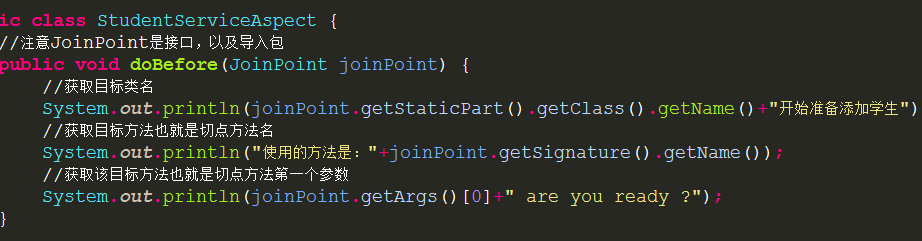
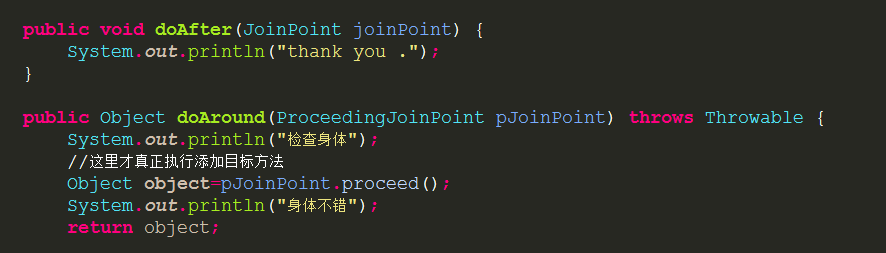
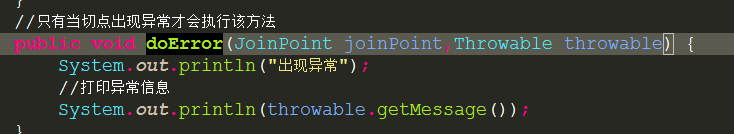
xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop

<http://www.springframework.org/schema/aop/spring-aop.xsd>">

上边的是beans.xml的头部配置

第四节Spring对Dao的支持 

首先要添加几个jar包，然后需要在<?xml>标签添加

xmlns:context=<http://www.springframework.org/schema/context>

http://www.springframework.org/schema/context

<http://www.springframework.org/schema/context/spring-context.xsd>

声明几个指定的bean

<!-- apache提供的dbcp数据库连接池,需要先添加jar包 -->

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">

<property name="driverClassName" value="${jdbc.driverClassName}"/>

<property name="url" value="${jdbc.url}"/>

<property name="username" value="${jdbc.username}"/>

<property name="password" value="${jdbc.password}"/>

</bean>

<!-- 加载指定的properties文件 -->

<context:property-placeholder location="jdbc.properties"/>

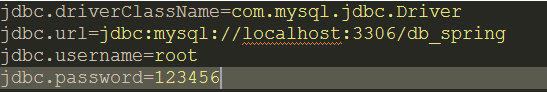
使用JdbcTemplate

<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">

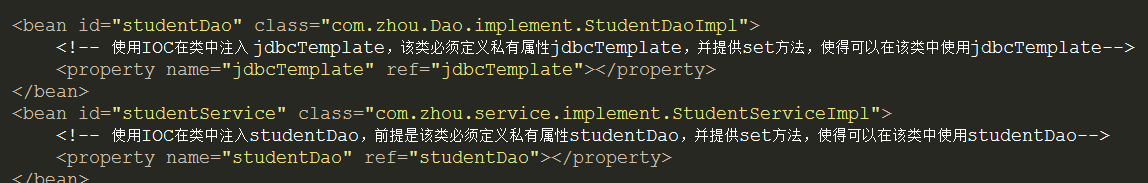
<property name="dataSource" ref="dataSource"></property>

</bean>

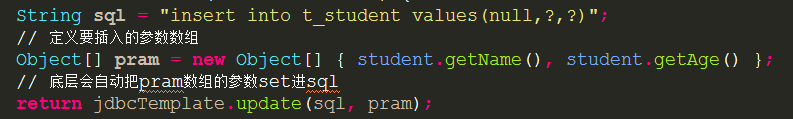
新建jdbc.properties文件，并配置必要的数据库连接信息

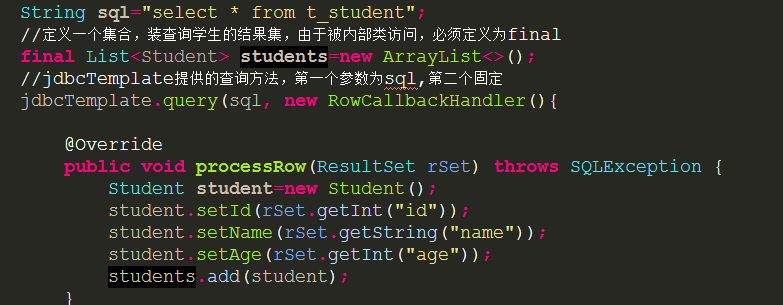


最后就是定义自己业务逻辑需要使用的bean

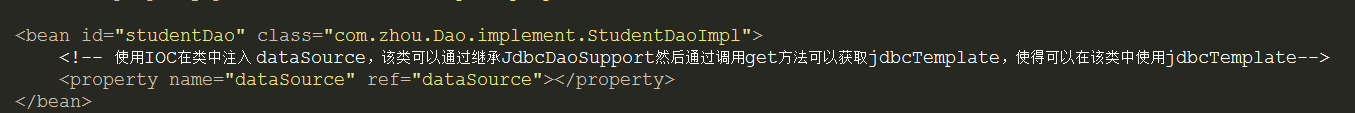


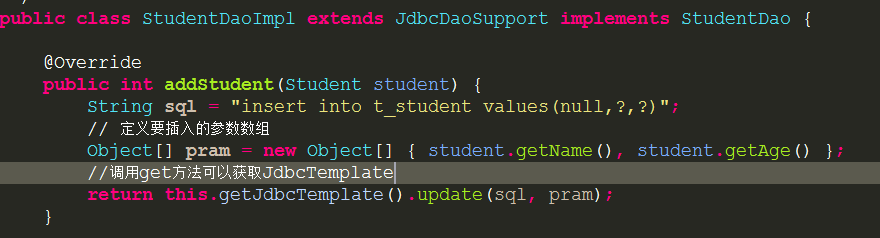
以及在业务逻辑中使用jdbcTemplate，进行数据库操作



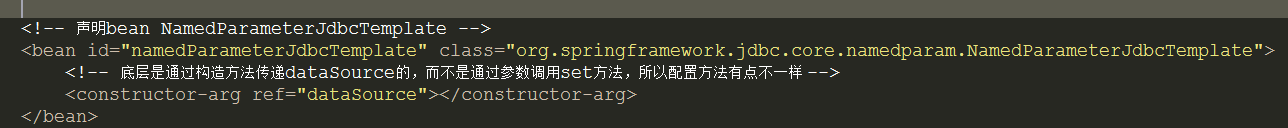


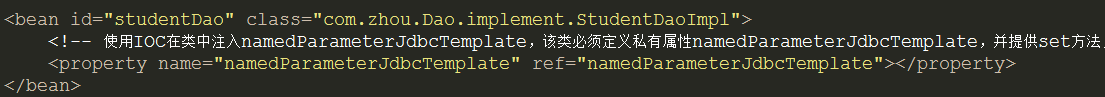
使用JdbcDaoSupport，比起其他两个省掉了一个声明bean的步骤

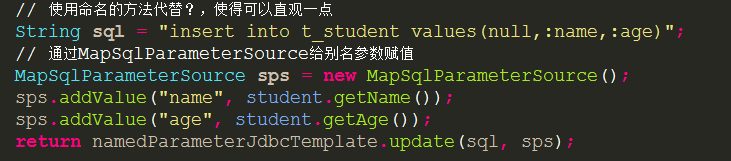


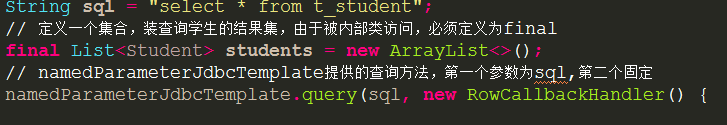


使用NamedParameterJdbcTemplate



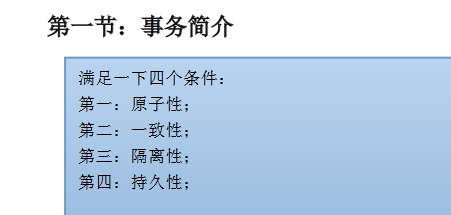


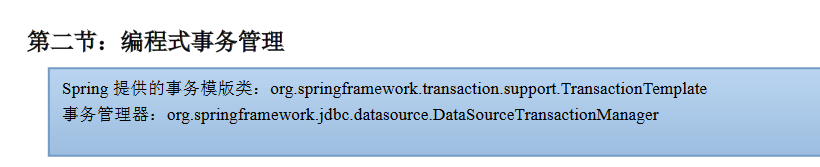




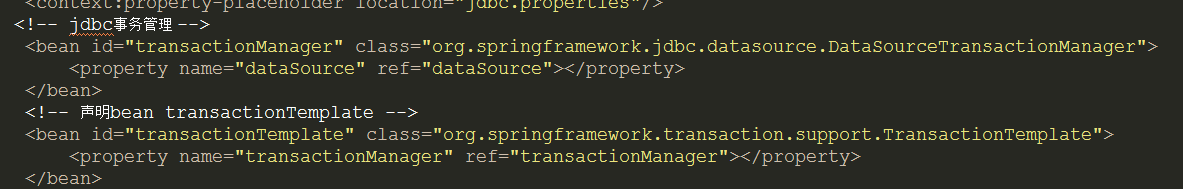
在Dao层具体实现数据库的增删改查操作，server层调用提供结果

Spring对事物的支持

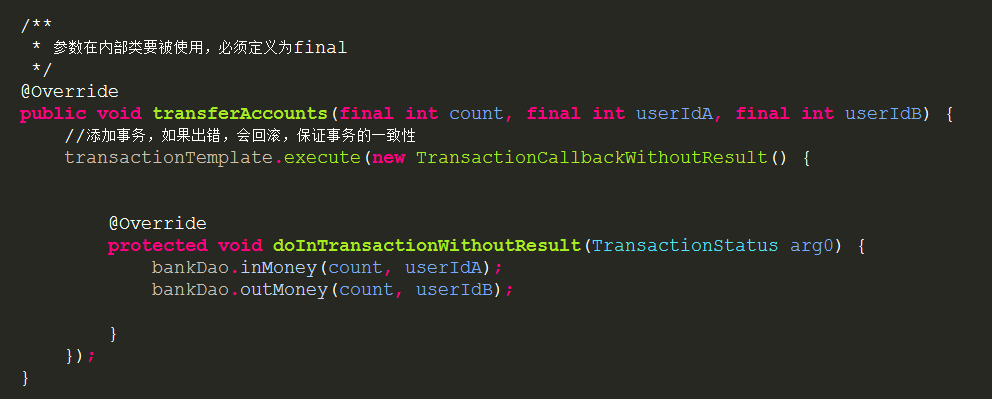
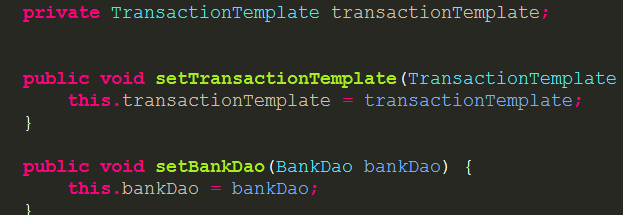
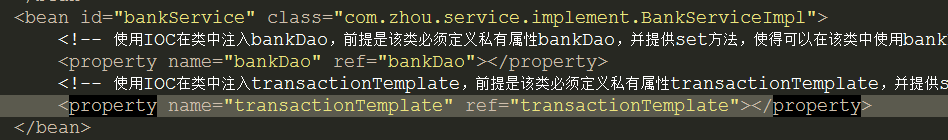




首先需要声明两个bean



然后在使用到事务的service层的类中注入transactionTemplate，并在该类中提供注入需要的set方法，以及使用transactionTemplate，进行事务处理



声明式事务管理

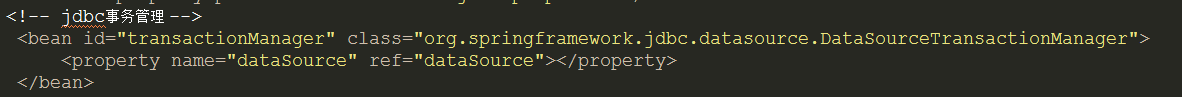
1. 使用xml配置声明式事务管理，添加依赖

xmlns:tx=<http://www.springframework.org/schema/tx>

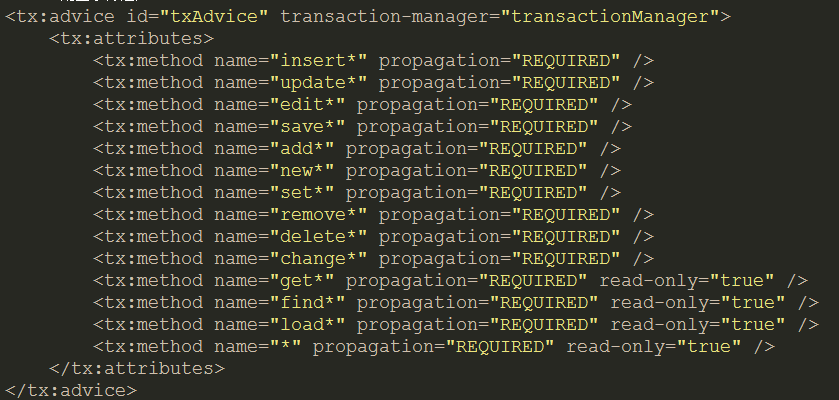
http://www.springframework.org/schema/tx

<http://www.springframework.org/schema/tx/spring-tx.xsd>

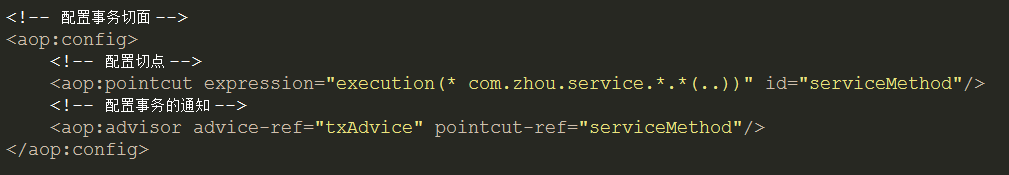
然后声明一个事务管理



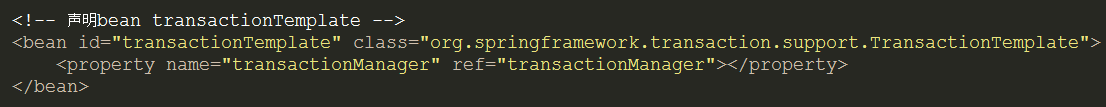
配置事务通知

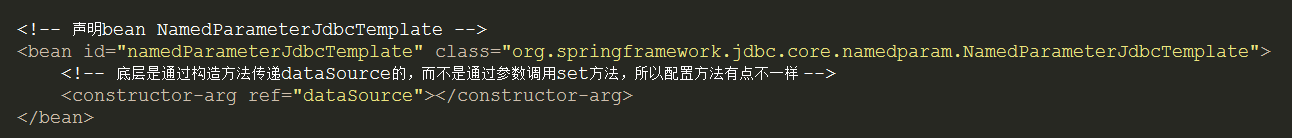


配置事务切面，也就是需要使用到事物的service层，不再需要使用注入



最后配置需要用到的各种bean，还有自己定义的类的bean





1. 使用注解式声明事务

只需在声明了事务管理之后加上下边的代码，以及配置自己需要用到的bean，然后在使用事务的类中配上注解

