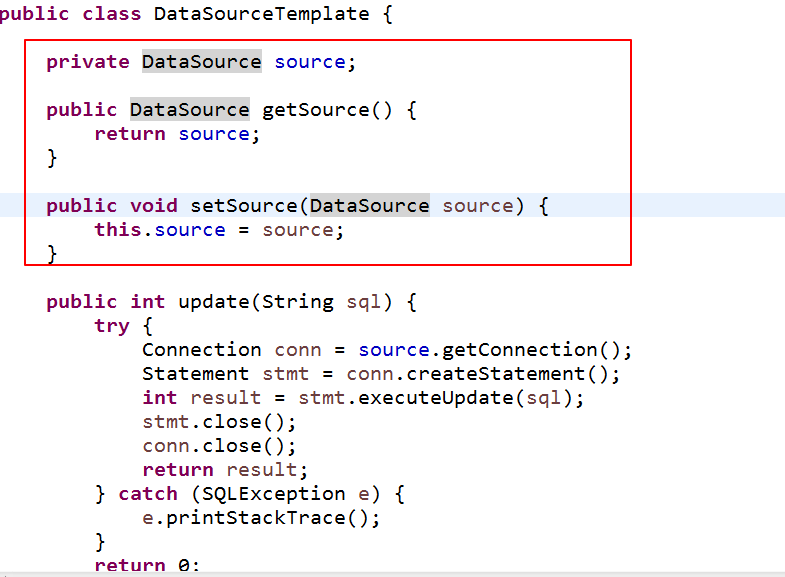
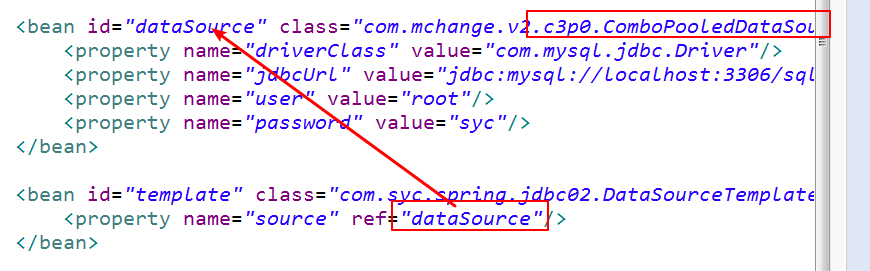
# Day15\_Spring\_JDBC模块

## 自定义DataSourceTemplate,模拟JdbcTemplae的实现

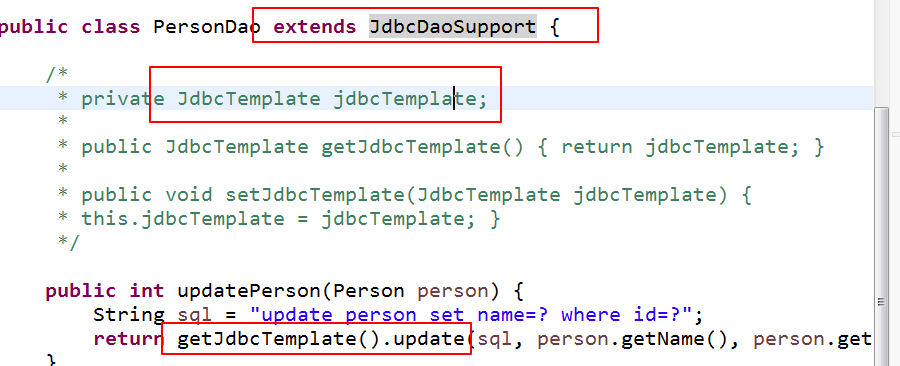


### 给DataSource属性注入值:

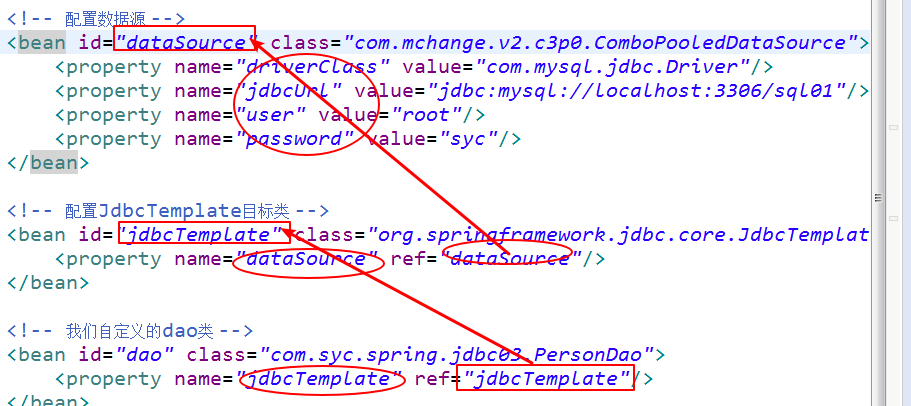


## JdbcDaoSupport类的使用

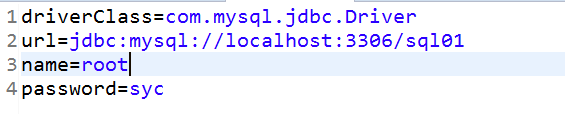
### 1.继承JdbcDaoSupport

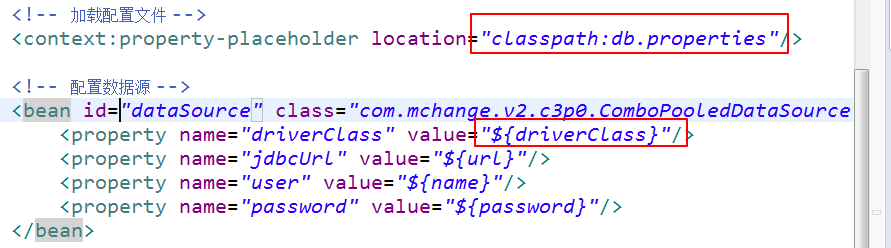


### 2.配置xml文件



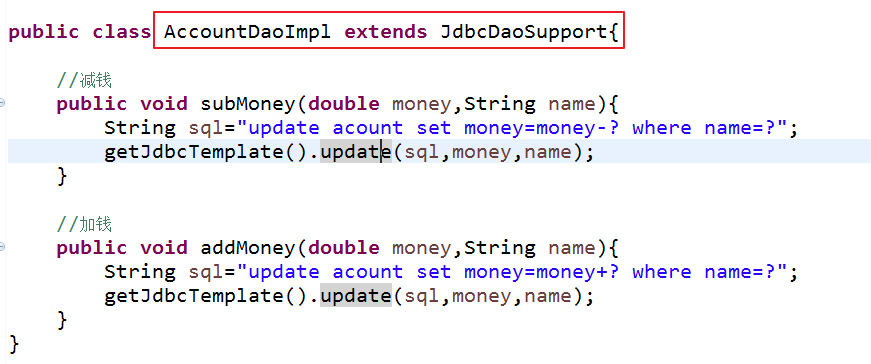
### 3.读取db.properties配置文件



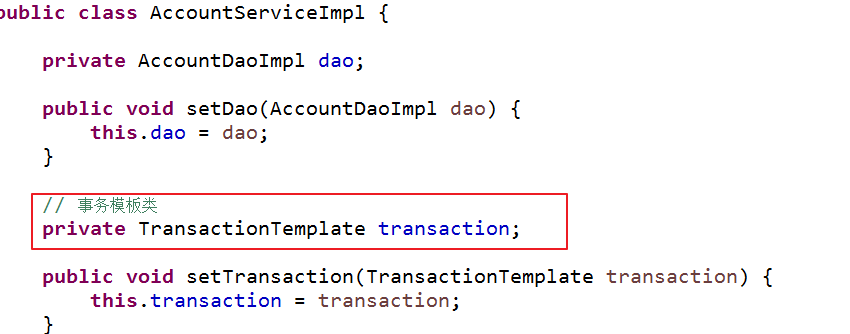


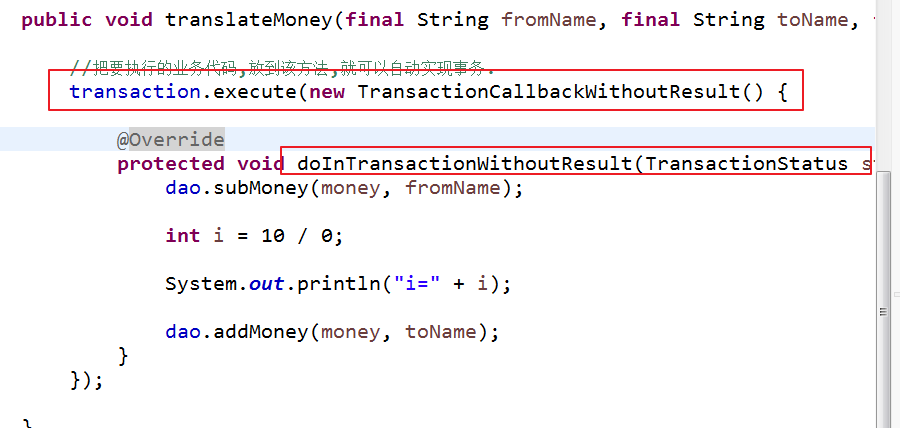
## Spring的数据库Transaction

### 1.创建Dao类

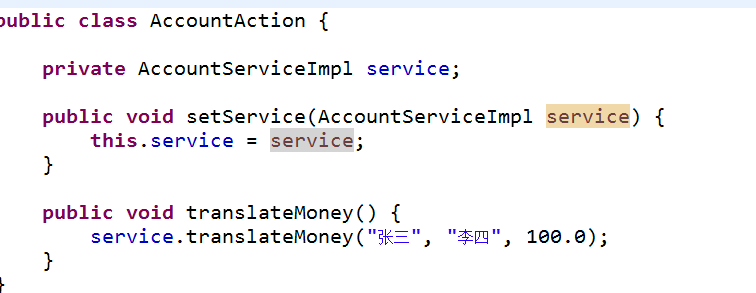


### 2.创建Service类

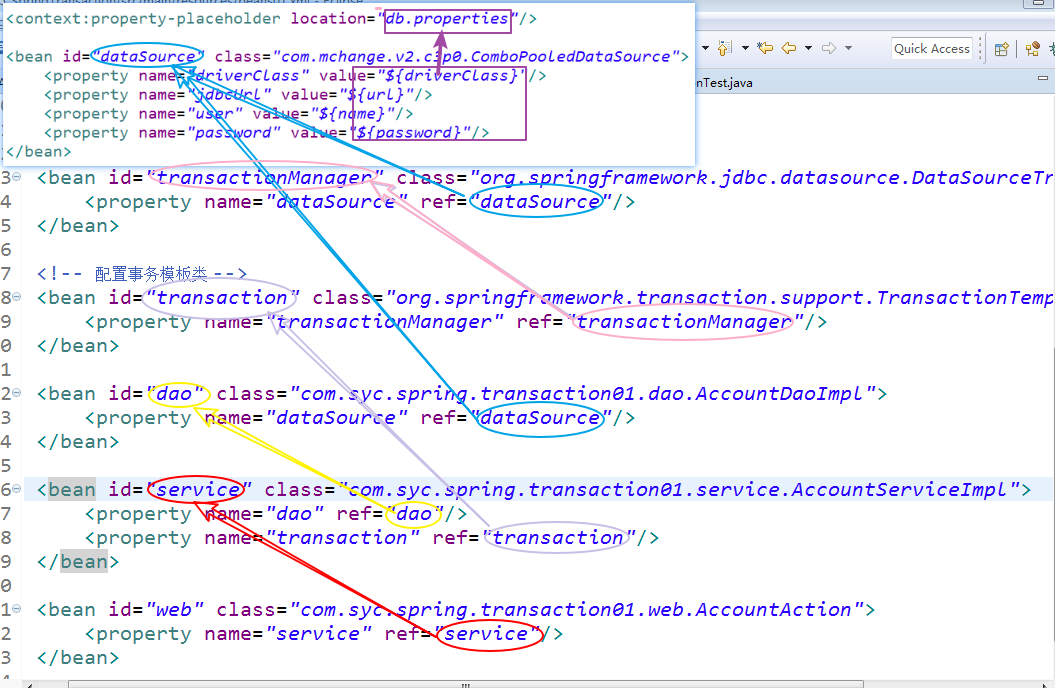




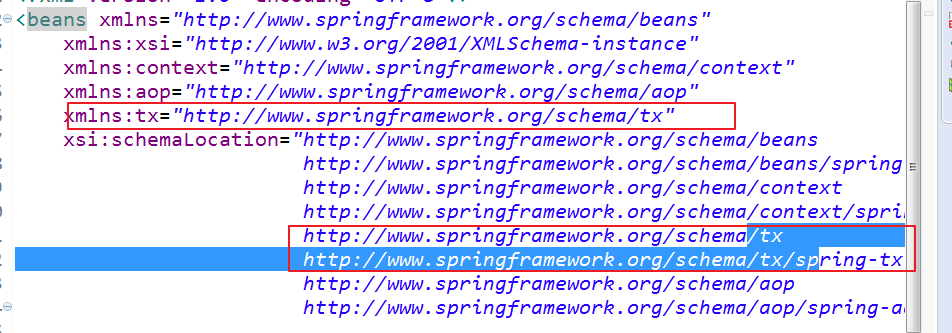
### 3.创建Web类

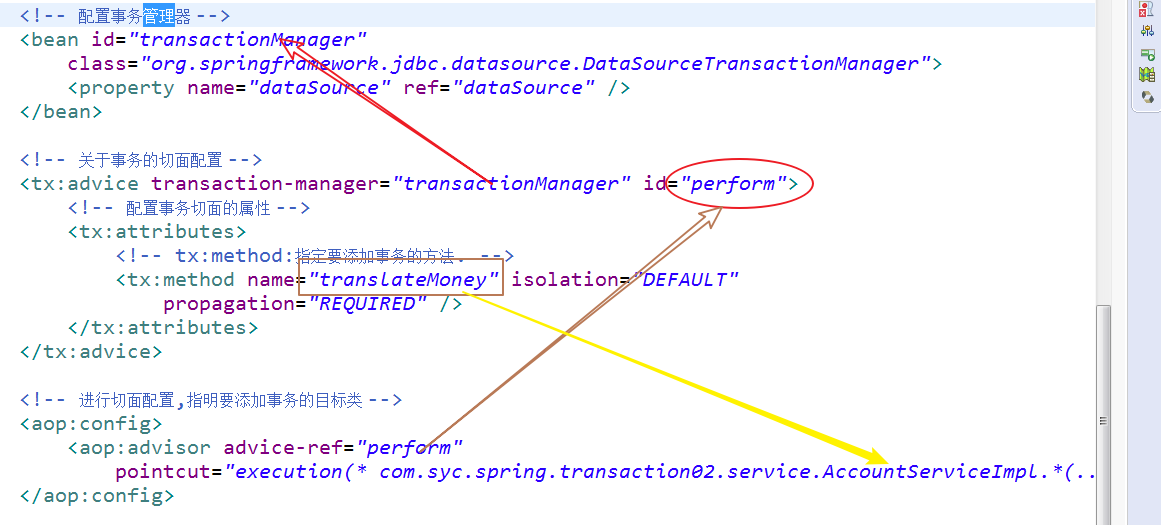


### 4.配置bean.xml文件



## 利用AOP实现事务操作





#### 事务的隔离级别

READ\_COMMITTED:读已提交

READ\_UNCOMMITTED:读未提交

REPEATABLE\_READ:重复读

SERIALIZABLE:可序列化

#### 事务的传播行为propagation:

REQUIRED:默认值,如果当前已有事务,则使用当前事务,否则新建一个事务.

MANDATORY: 如果当前已有事务,则使用当前事务,但是如果没有事务,则抛出异常.

NESTED:支持事务的嵌套,可以在嵌套的事务中执行.

NEVER:不支持当前事务,如果存在事务则会抛出异常.

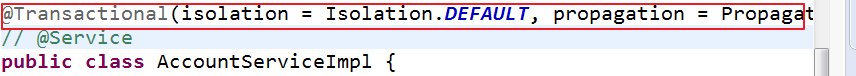
NOT\_SUPPORTED:不支持当前事务,总是在无事务状态下运行.

REQUIRES\_NEW:不管当前是否已存在事务,都会新建一个事务来运行.

SUPPORTS:如果存在事务,则使用当前事务,否则就在无事务状态下运行.

## 注解方式实现事务

### 给目标类添加@Transactional注解



### 配置<tx:annoation-driven>注解驱动器



#### 思考:

#### 注解方式中,Spring是如何给目标类自动开启事务的?