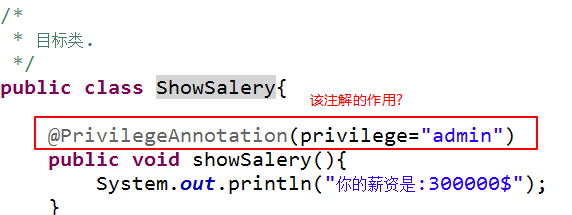
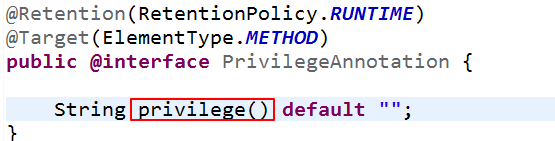
# Day14\_Spring\_AOP注解的实现方式

## 薪资查看系统总结

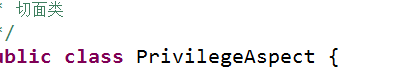
### 1.创建目标类

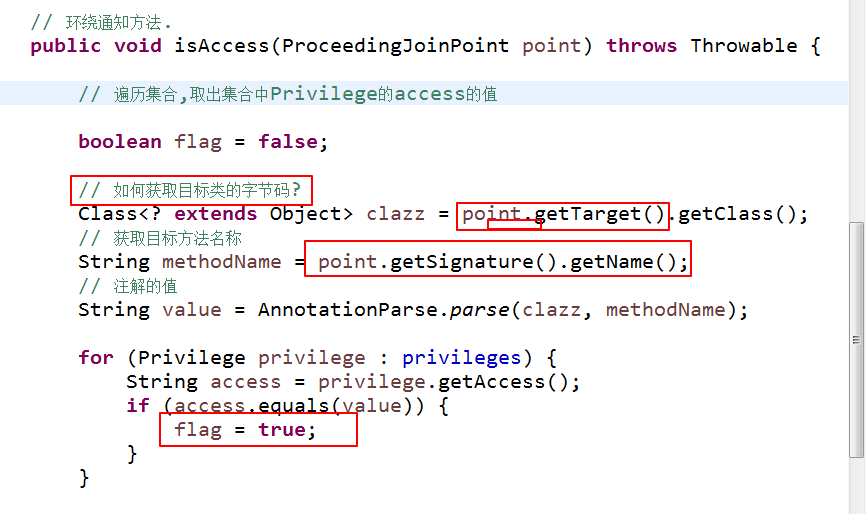


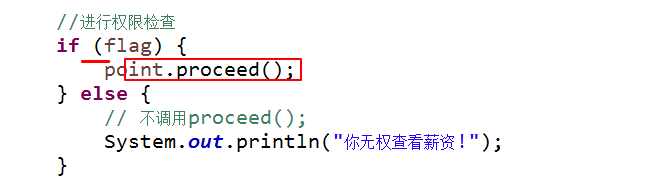
### 2.自定义注解



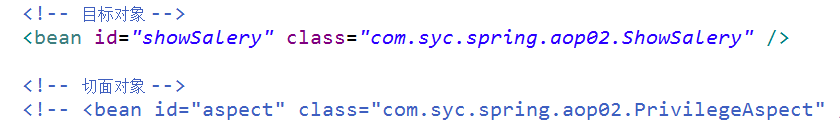
### 3.创建切面类

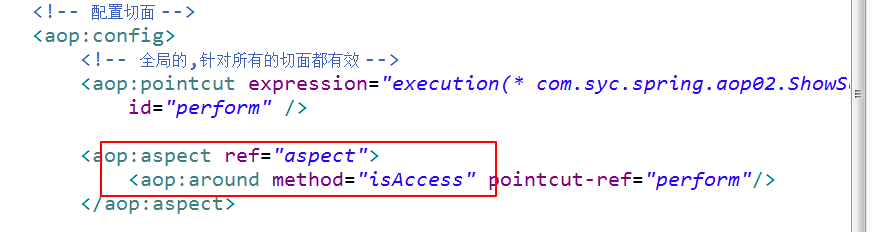




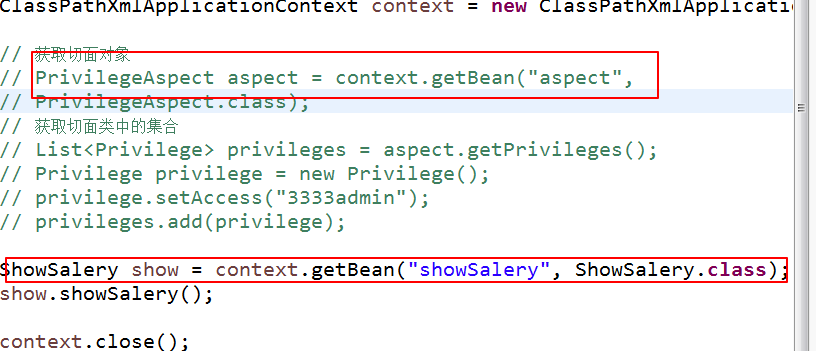


### 4.配置bean.xml文件,配置切面和目标类

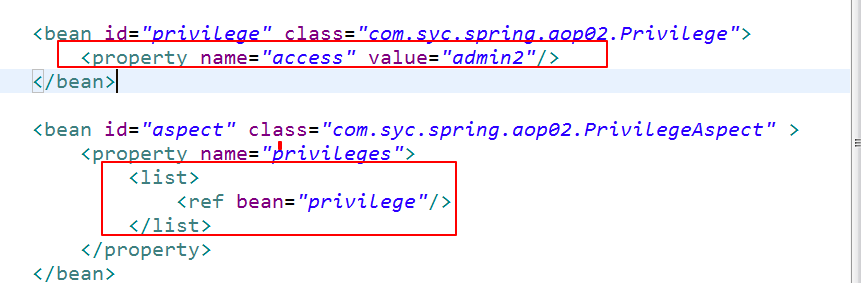




### 5.测试赋初值

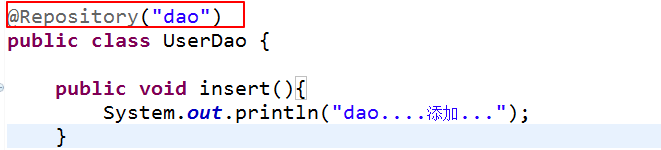


### 6.进行改造,把属性值的注入在xml中完成

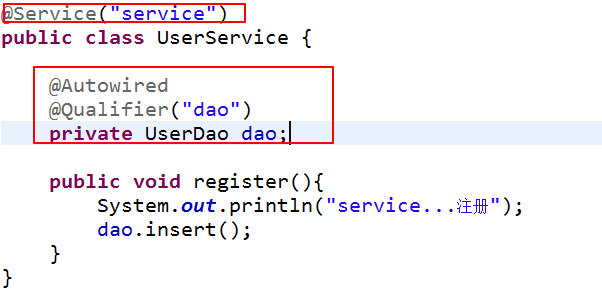


## 注解方式实现AOP

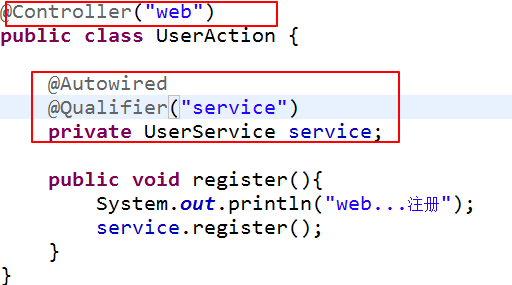
### 1.定义dao层类



### 2.定义service层



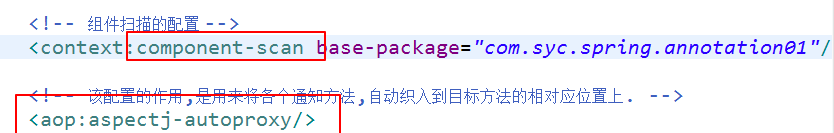
### 3.定义web代码

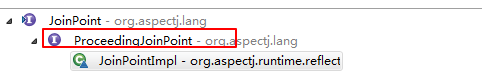


### 4.定义切面类



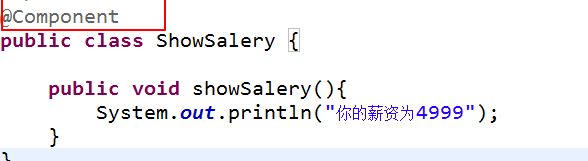
### 5.配置xml文件的相关属性





## 注解方式实现多切面配置

### 1.创建目标类



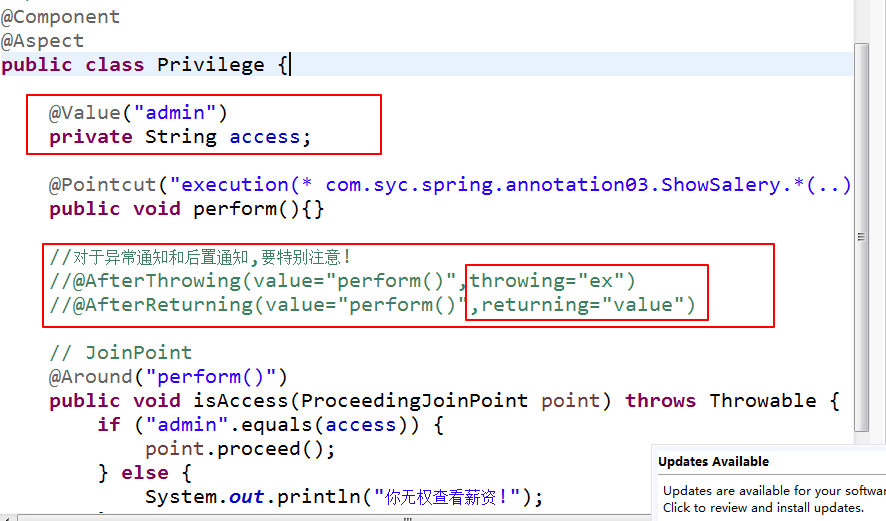
### 2.创建切面类一



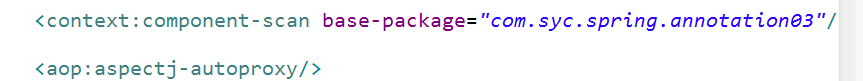
### 3.创建切面类二



### 4.创建切面类三



### 5.配置xml文件



### 6.通知的执行顺序总结(重点)

#### .在xml中通过aop:aspect方式配置:

此时执行的顺序默认情况下,与切面的注册顺序有关系,也就是谁先注册谁就先执行.但是我们可以在该aspect中,添加order属性,来改变通知的执行顺序.

order的值越小,优先级越高!

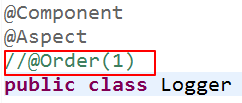


#### .以注解的方式配置aop切面

默认情况下,切面类的执行顺序,与该类的首字母顺序有关,也就是首字母越靠前,就越先执行.

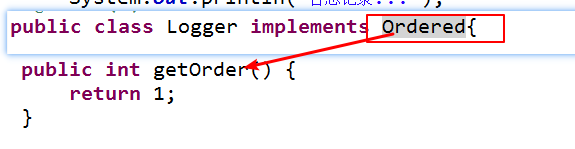
非默认情况下,可以通过@Order注解来改变,也可以通过让切面类实现Ordered接口,复写getOrder()方法来改变顺序.

##### @Order注解:





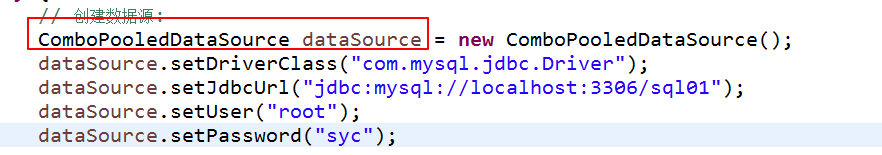
##### 实现接口:



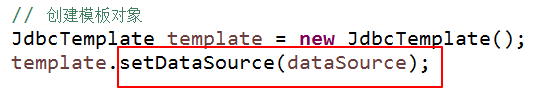
## Spring的JDBC模块

### JdbcTemplate,简化了原生的Jdbc的操作.

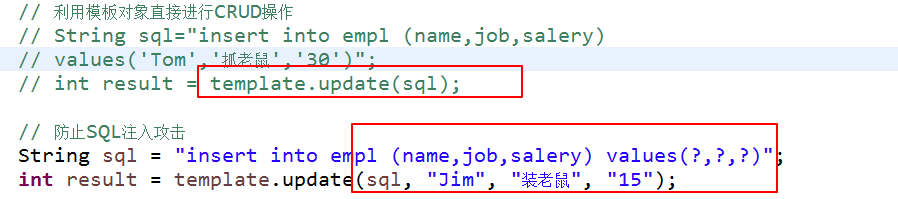
#### 创建数据源对象



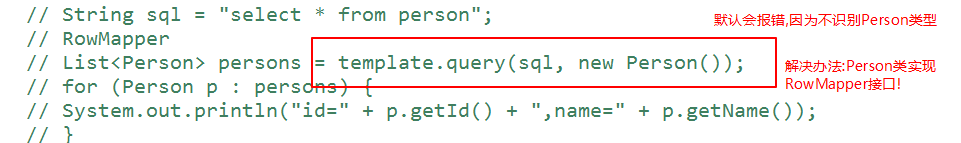
#### 关联数据源

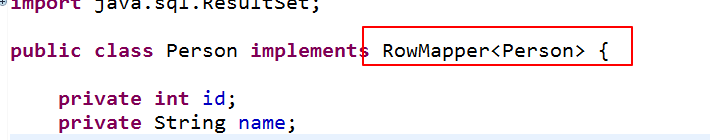


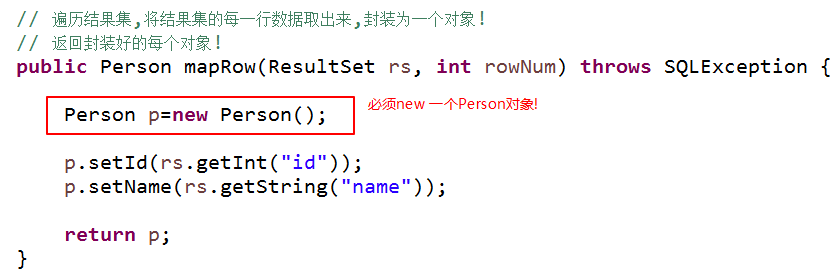
#### 进行CRUD操作



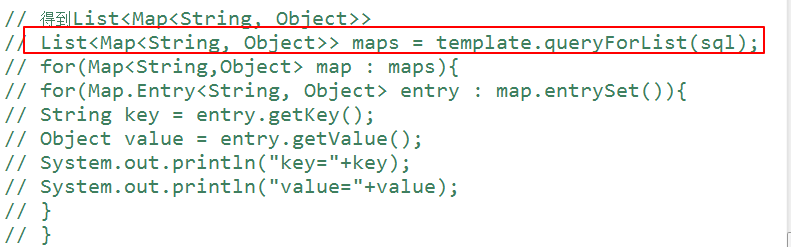
#### 实现查询操作







#### 查询得到Map集合



#### 注意:

