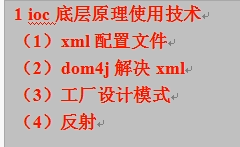
# Spring核心包括ioc aop两大部分 开发思想高内聚低耦合

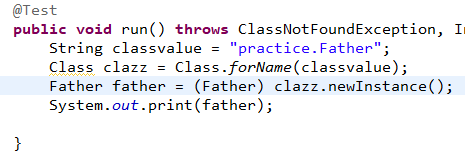
# aop 面向切面编程

## ioc 控制反转 对象创建交给spring

ioc实现原理



反射创建类代码



ioc简单实例

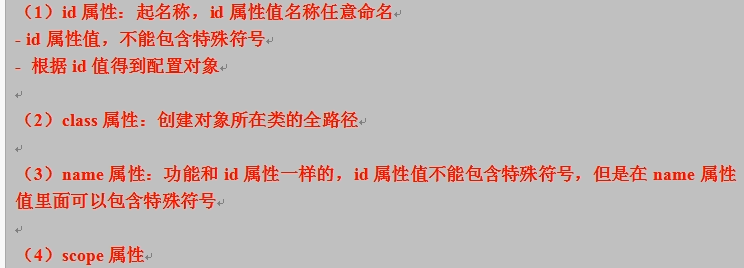
bean.xml



ioc操作通过配置文件或者注解两种方式

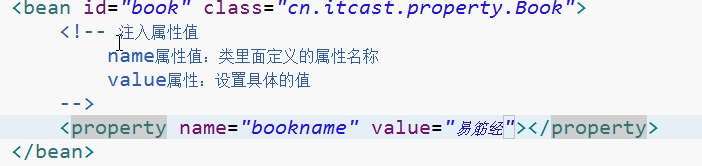


## bean标签常用属性

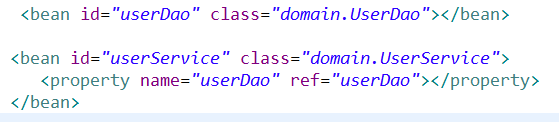




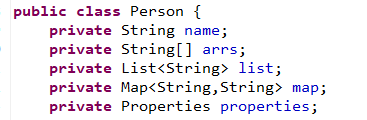
set方法注入属性



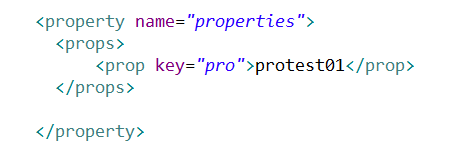
注入对象属性



注入复杂属性(数组 list map…)



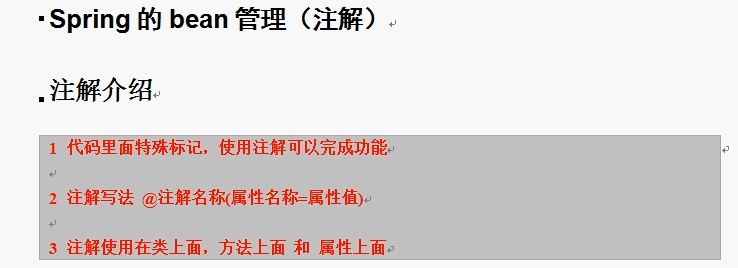




p名称空间



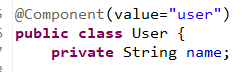


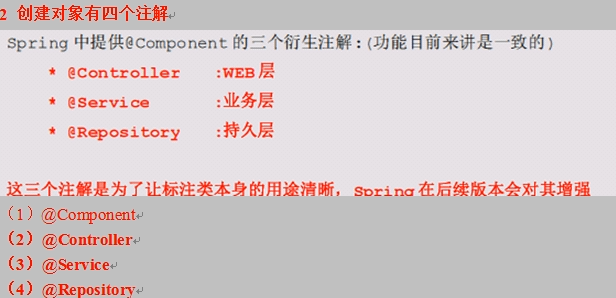


配置文件扫描bean存放的包 如果多个包用相同的前缀名即可



类注解



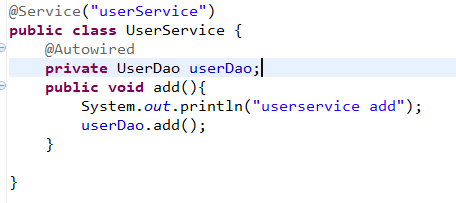


注解注入属性

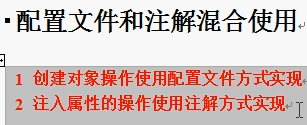
(Autowired会按照类名寻找响应类 会自动生成set方法

Resource则按照对应的id寻找相应类

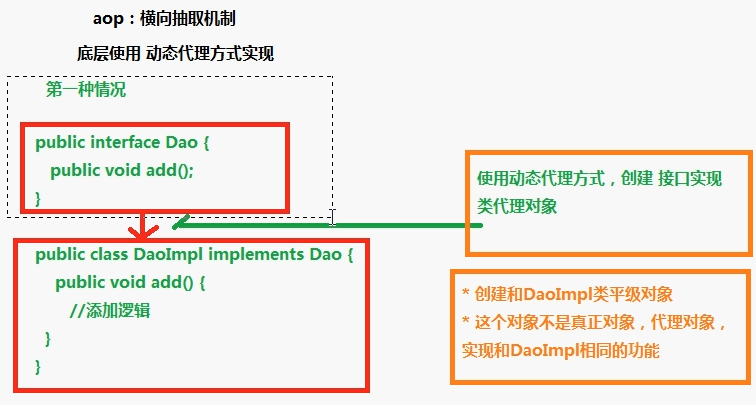
)







# aop 面向切面编程,拓展功能不修改源代码



# 操作术语：

连接点：类中可以被增强的方法。

切入点：类中有很多方法可以被增强，实际操作中被增强的方法才叫切入点。

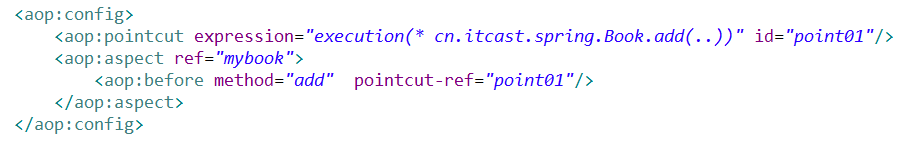
通知/增强：实际被增强的逻辑 例如扩展日志功能（前置 后置 异常 最终 环绕）。

切面：把增强应用到具体方法上，这个过程称为切面。

spring中aop操作通过aspectj实现

aspectj不是spring的内容 是第三方的技术

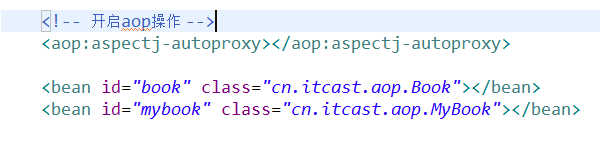
实现方式有xml配置和注解两种方式



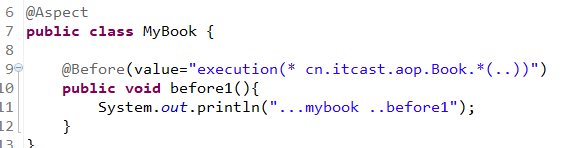
注解方式

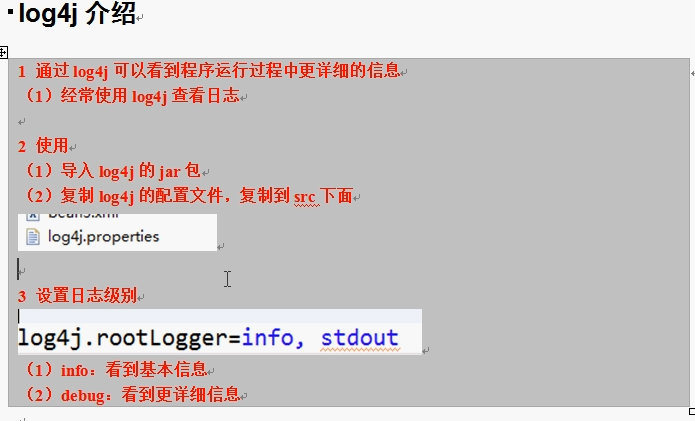
创建对象

配置文件开启注解扫描

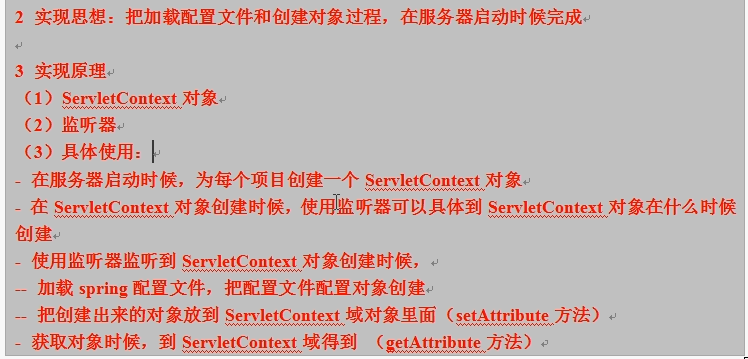


在增强类上配置注解信息



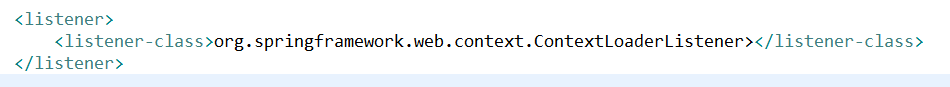


## Spring整合web项目



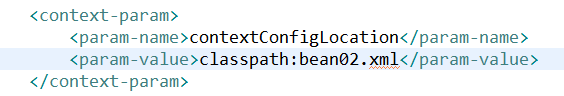
实际项目中spring帮我们封装好了 只需要我们配置监听器

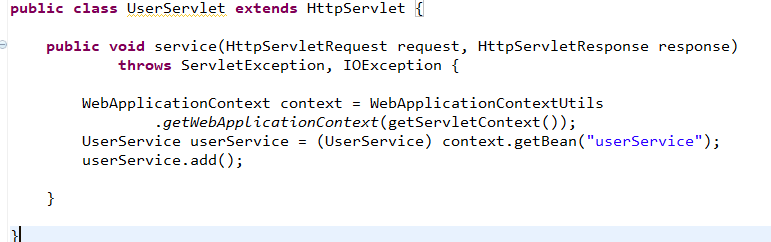
配置监听器



org.springframework.web.context.ContextLoaderListener

指定spring配置文件的位置



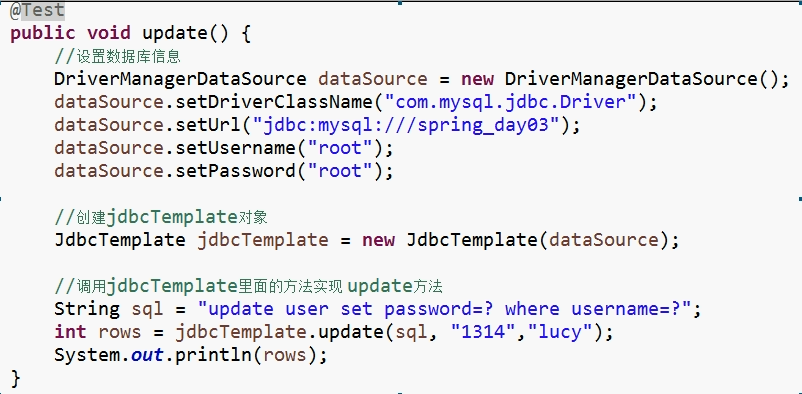


## JdbcTemplate操作

添加

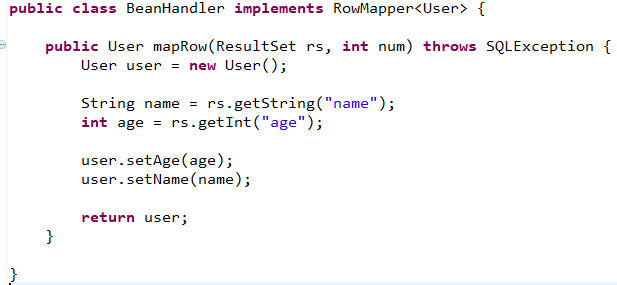


修改

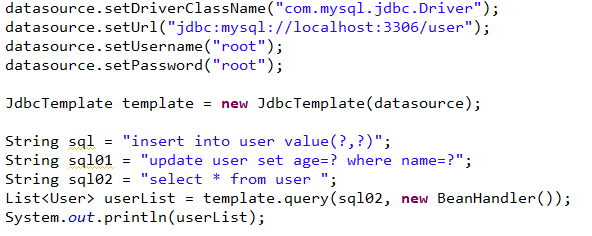


查询

JdbcTemplate查询接口没有实现类 需要手动编写



查询list



### Spring配置c3p0连接池

在配置文件中配置c3p0连接

