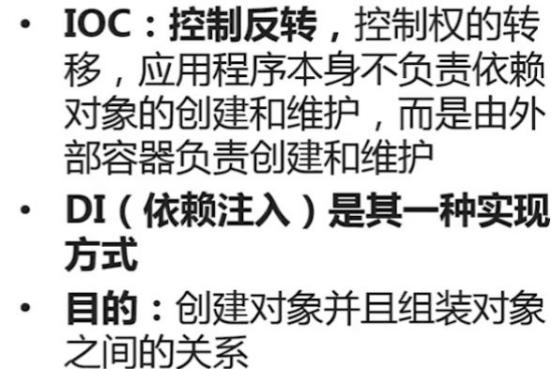
1:

由容器动态的将某种依赖关系注入到组件之中

2：



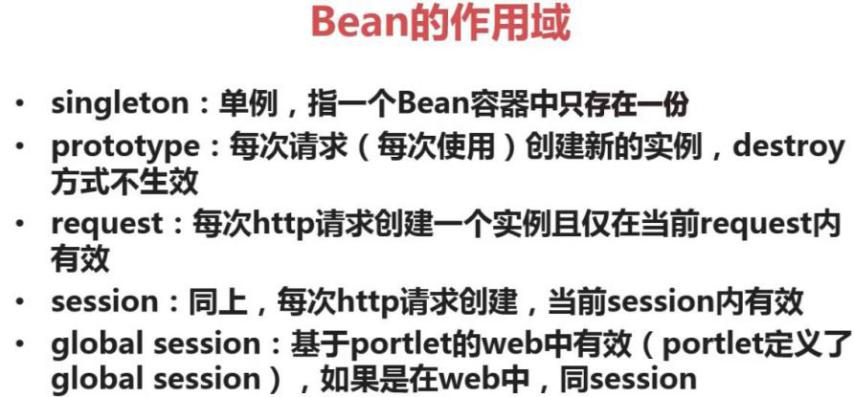


相关代码见：springTest工程springTest1包。

3：

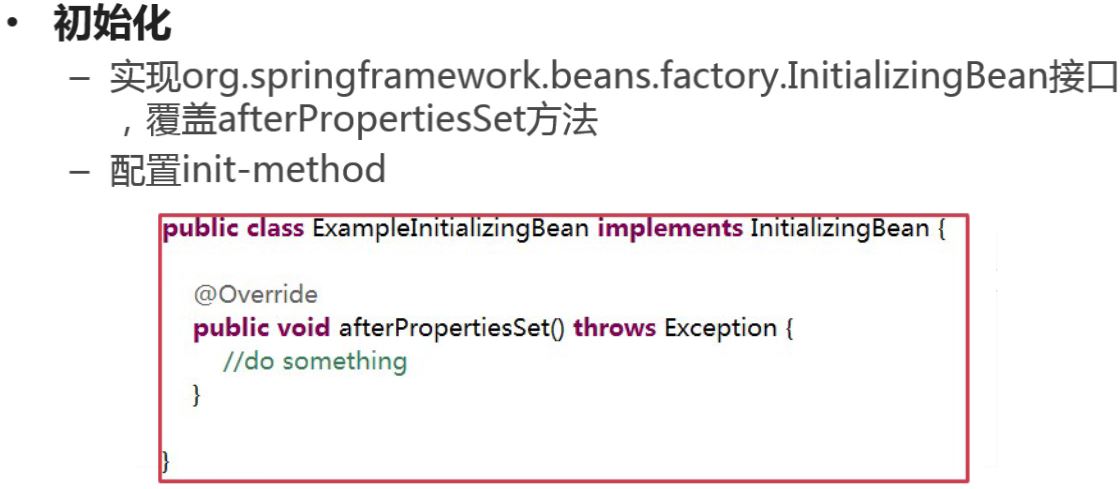


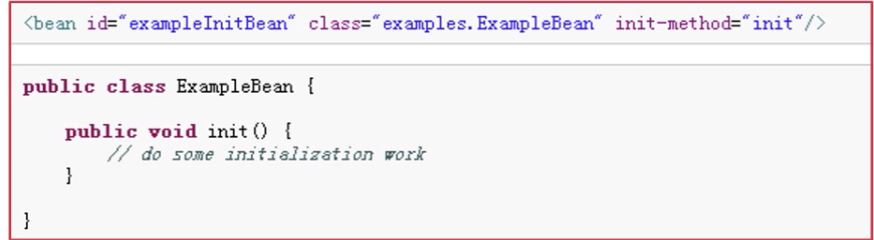
Class是要实例化的类；Scope是范围，作用域；Autowiring mode自动装配的模式；Initialization/destruction method初始化和销毁方法。

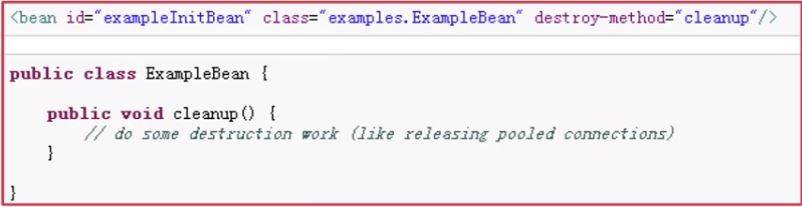
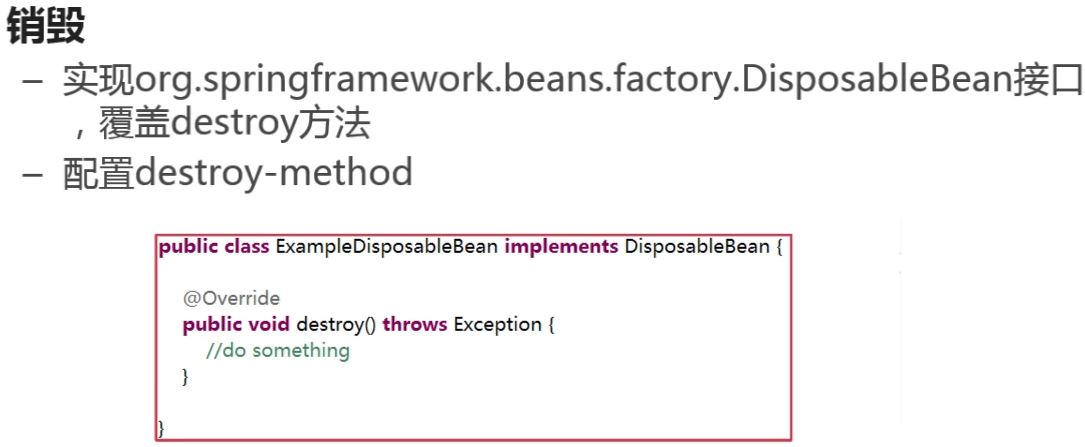


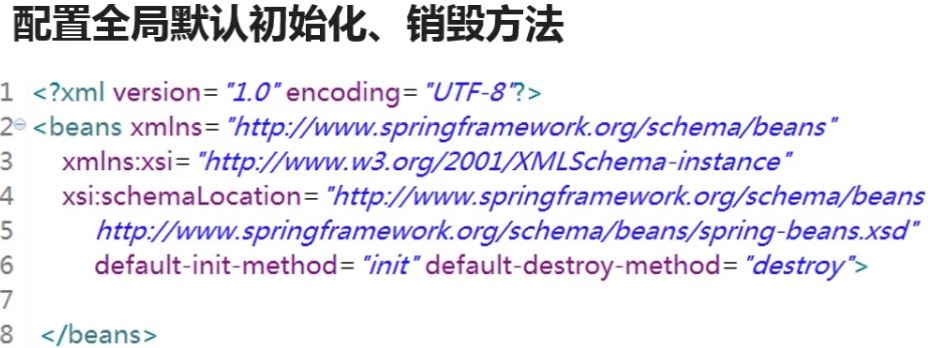
代码见springTest工程springTest2包。

4：

Bean的

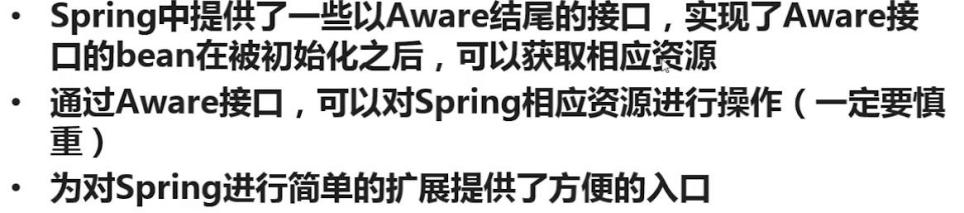






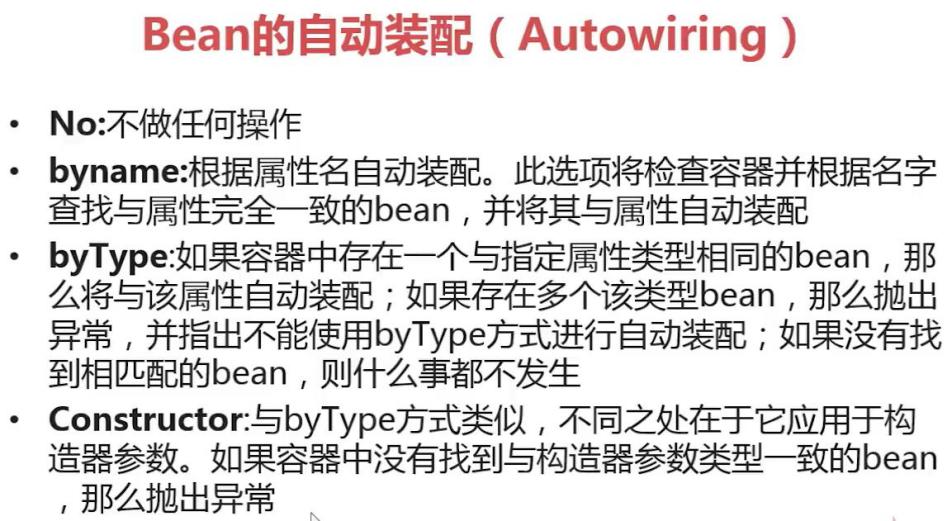
当三种方法同时从在时，默认的则不执行，接口初始化销毁和配置文件初始化销毁会执行，且接口方法先于配置文件执行。代码见springTest工程springTest3包。

5：



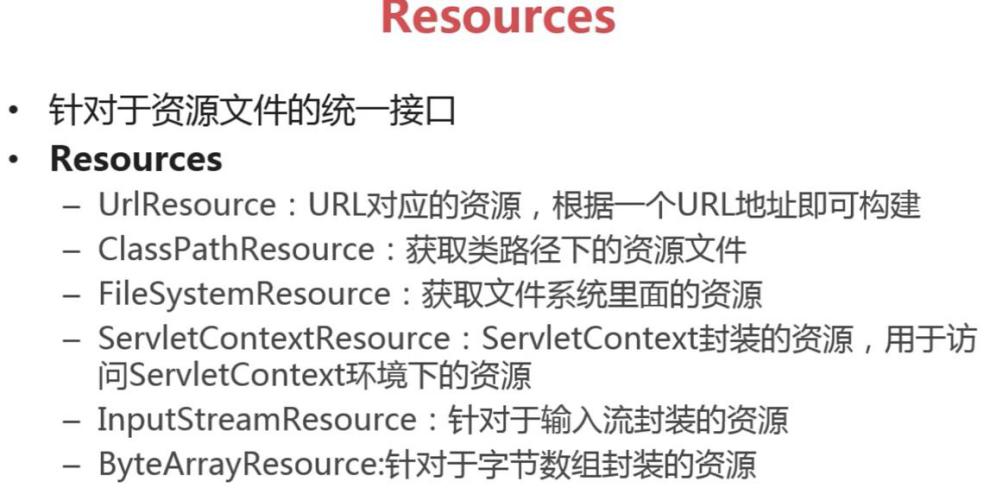
Aware结尾的一下接口：代码见springTest工程springTest4包。

6：

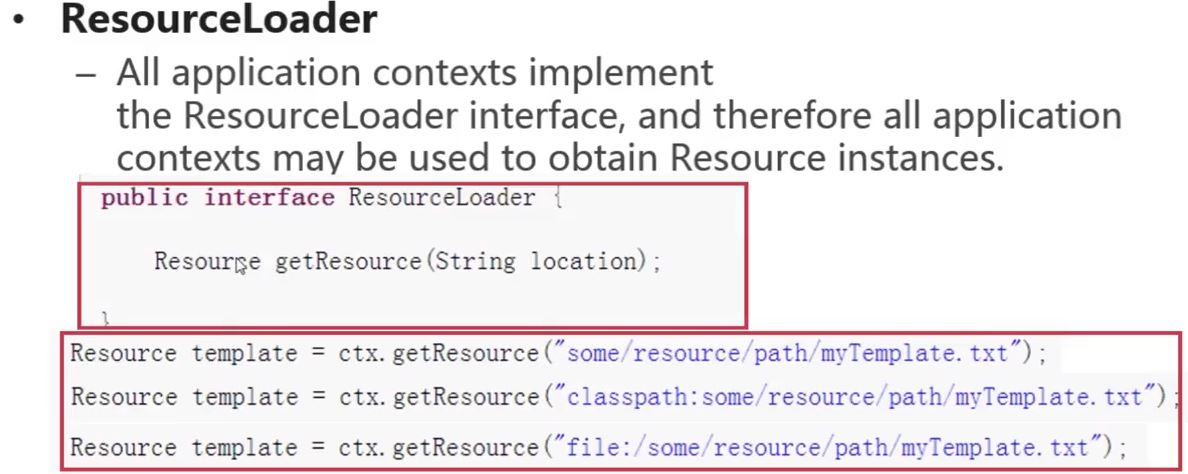


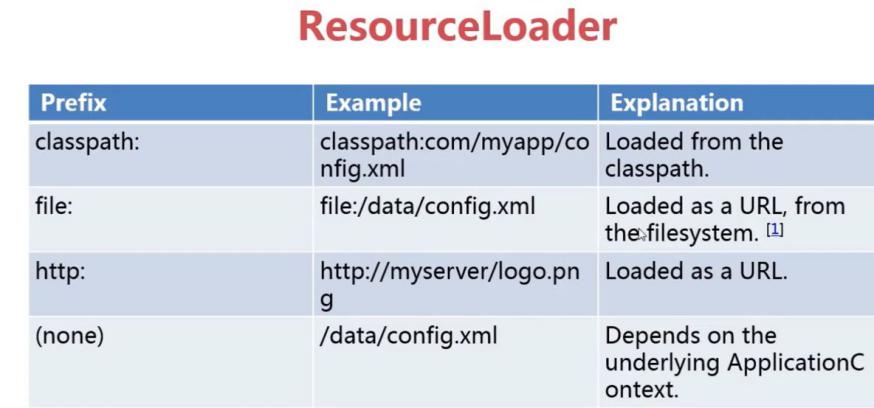
用byName的时候，bean id=“xxx”一定要与setXxx（形参）的Xxx首字母变小写后相同，而不一定与形参相同，也不一定与成员变量相同。当然，一般正规的set方法后面的Xxx是与成员变量相同的，因此一般情况下bean id与成员变量相同。Constructor自动装配是根据bean的类型来装配的。代码见springTest工程springTest5包。

7：



Spring中的自动装配有哪些限制？  
答：  
- 如果使用了构造器注入或者setter注入，那么将覆盖自动装配的依赖关系。  
- 基本数据类型的值、字符串字面量、类字面量无法使用自动装配来注入。  
- 优先考虑使用显式的装配来进行更精确的依赖注入而不是使用自动装配。



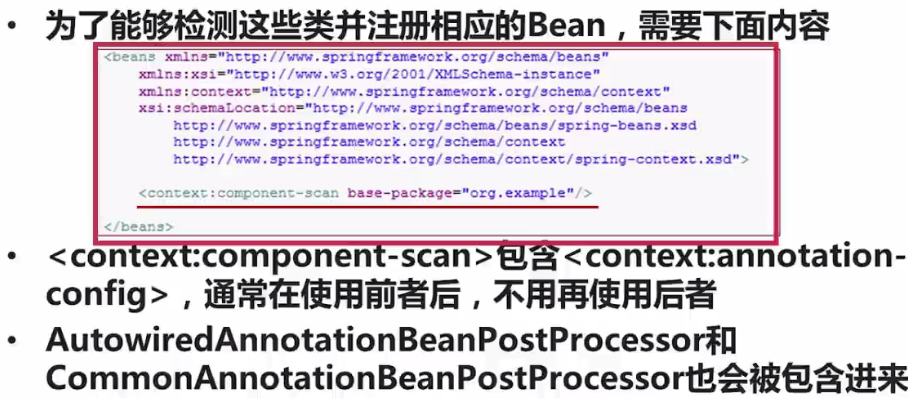


相关用法及代码见：springTest工程springTest6包。

8：通过注解装配bean：能实现自动参数的注入：

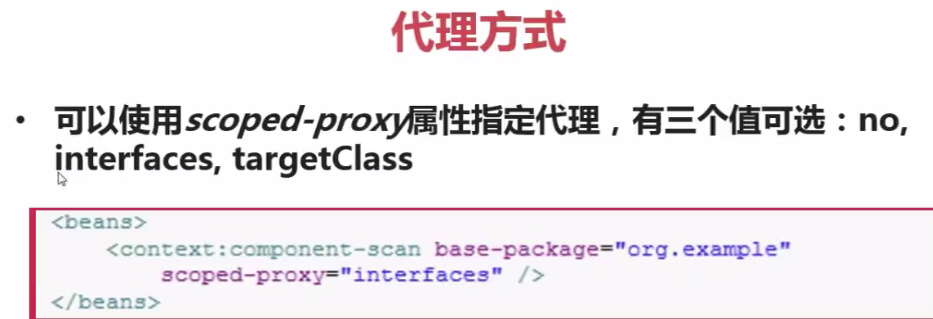






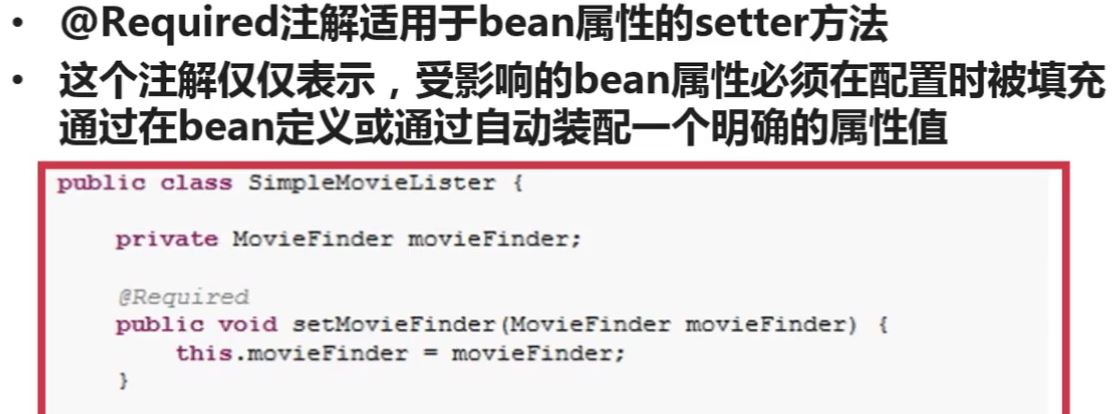






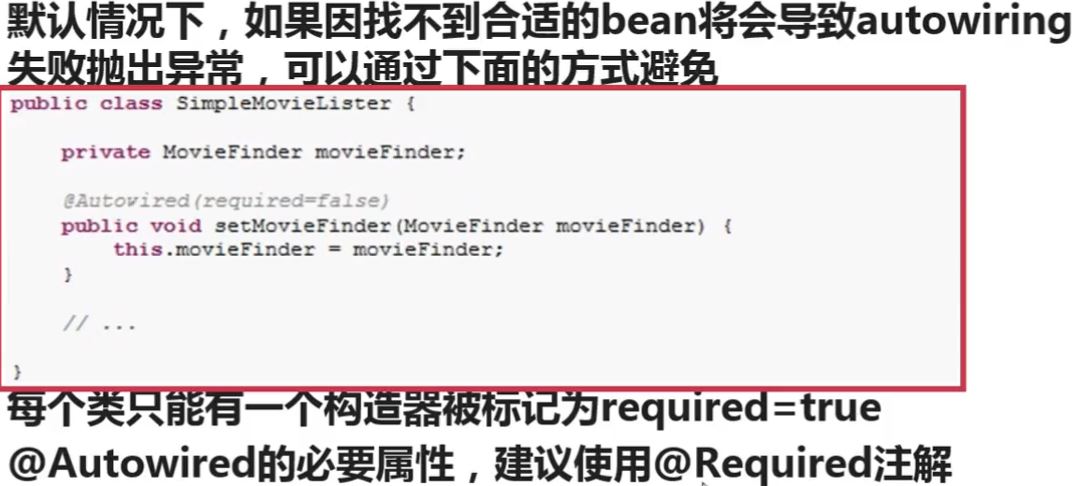
相关用法及代码见：springTest工程springTest7包。

9：





@Autowired更常用。





Required=false表示非必须的。@Autowired注解在set方法，构造器或是成员变量上作用都是一样的。相关用法及代码见：springTest工程springTest8包。

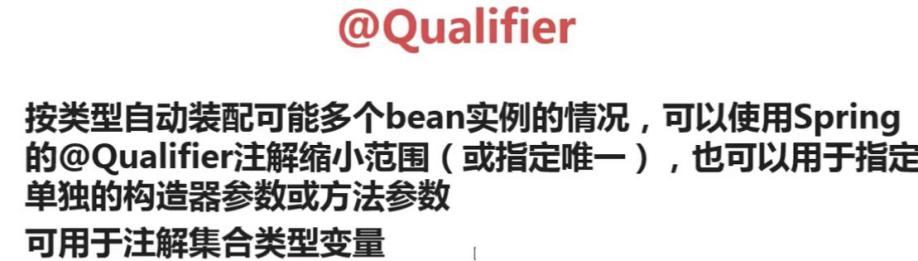


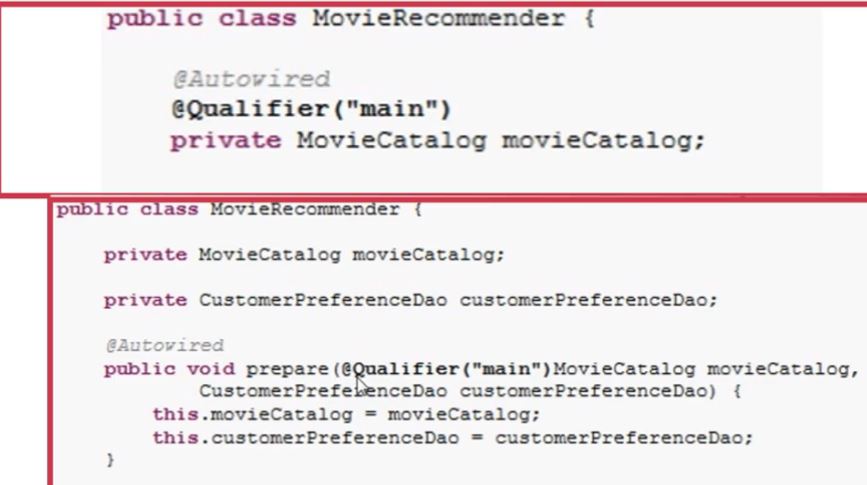
意思是可以给set，list等集合添加注解，包括他们的内部成员变量。@order对map无效。

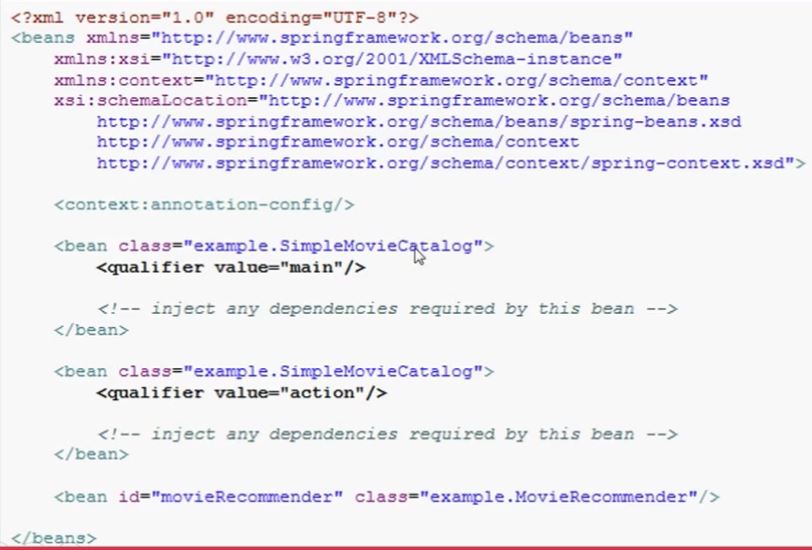


相关用法及代码见：springTest工程springTest9包。

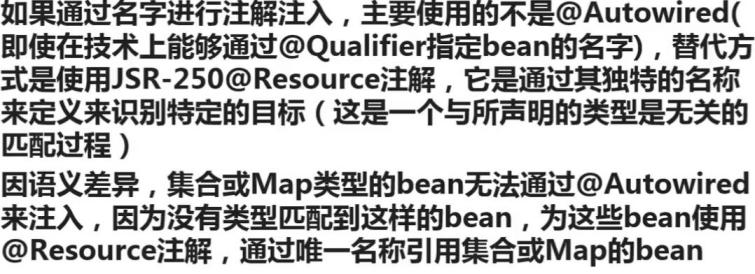
10：

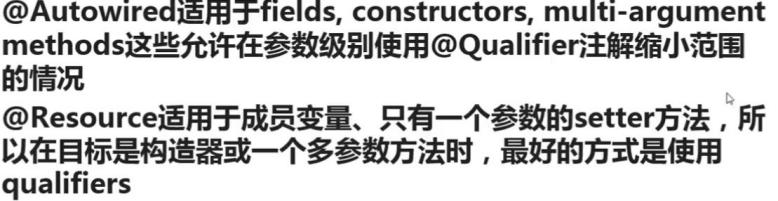




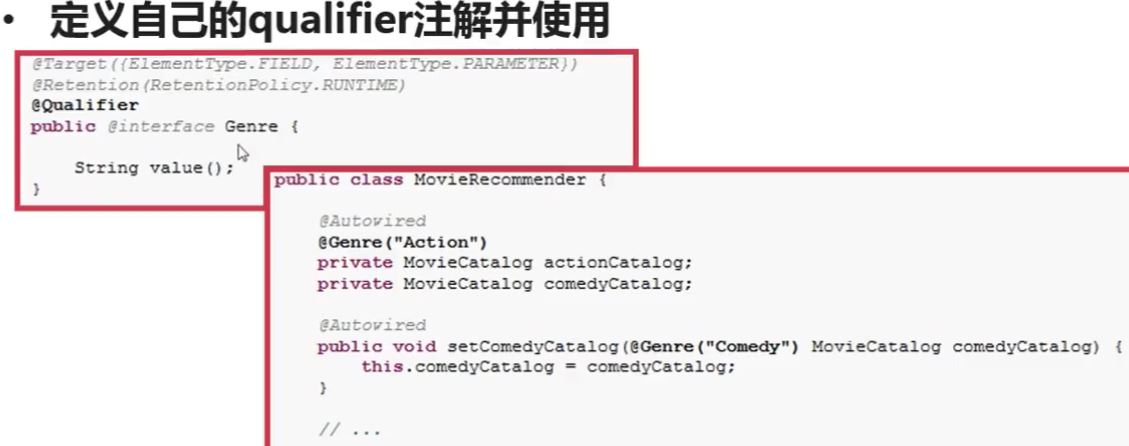


Qualifier在配置文件中的配置。

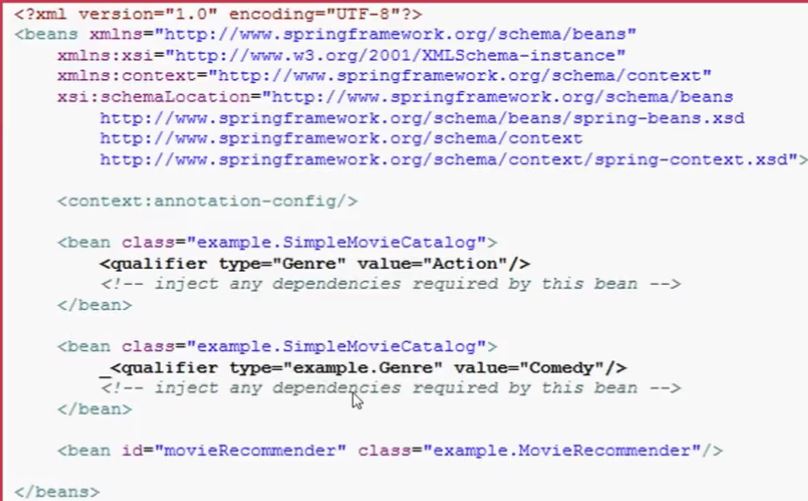




Fields字段，constructor构造器，multi-argument methods多参数的方法。



用qualifier来注解自定义的注解。则自定义的注解就有了qualifier的相关功能。



使用配置文件自定义注解。相关用法及代码见：springTest工程springTest9包。

11:



这两种方法效果是一样的，都是向IOC容器中添加一个对象。

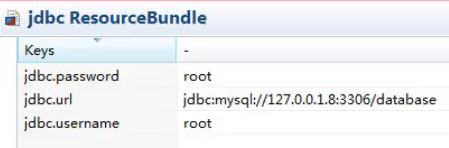


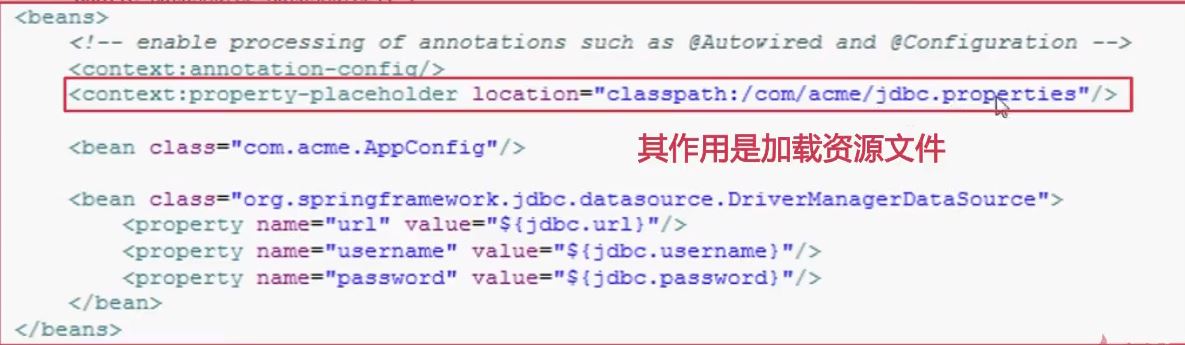
proxyMode是指使用代理。

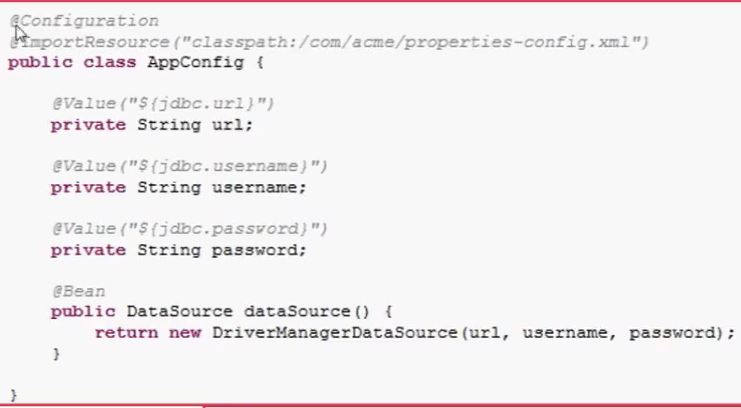


@Bean没有指定名字则默认为方法名。相关用法及代码见：springTest工程springTest10包。

12:

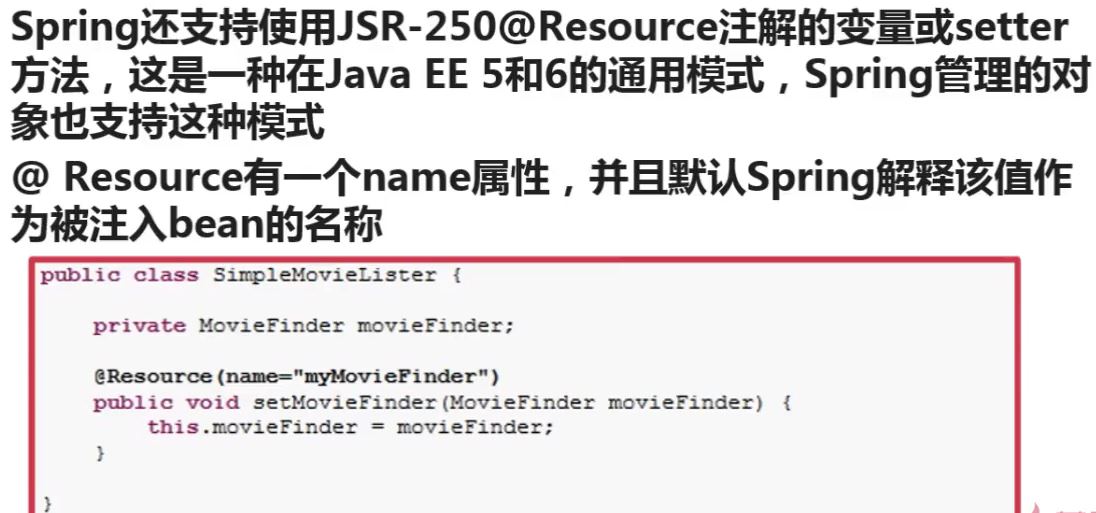




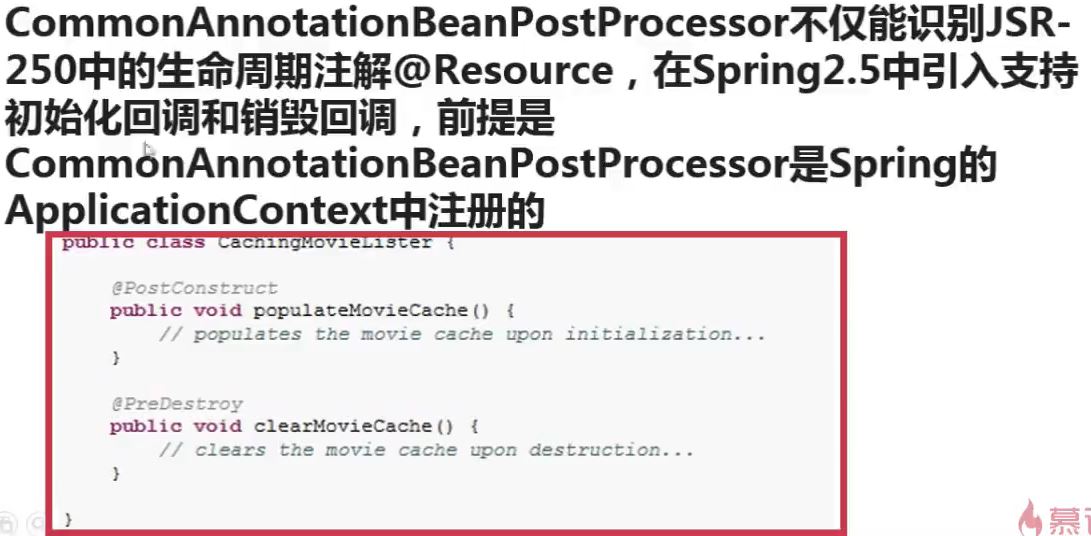


@ImportResource跟配置文件，配置文件中配置读取的文件，@Value跟文件中的名字来获取他所代表的值。相关用法及代码见：springTest工程springTest10包。

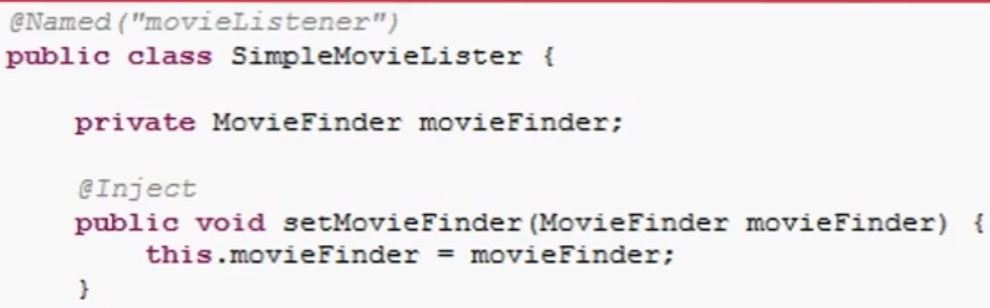
13：







@PostConstruct用在初始化方法，@PreDestroy用在销毁方法。



相关用法及代码见：springTest工程springTest11包。

@Autowired 默认by type，失败才by name；而@Resource 刚好相反。

