Spring的IoC创建Bean时的一些细节

# Spring创建Bean对象时

## 单例/多例（作用范围scope）

通过scope配置：不指定或指定scope=”singleton”，就是单例模式。获取的该类的对象是同一个。scope=”prototype”多例。



## 什么时候创建Bean对象



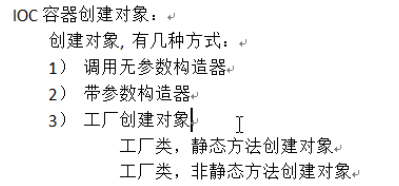
对于作用范围为**scope=”singleton”**的bean对象，默认容器初始化时就会创建，如果不想容器初始化时被创建，利用**lazy-init="false"**声明。（**懒加载**）默认为true。默认为false，不延迟创建。只对**scope=”singleton”**单例有效，因为多例肯定是需要的时候才创建。



## 指定init与destory方法：（属于bean的生命周期方法）



# Spring IoC容器创建Bean对象的方法



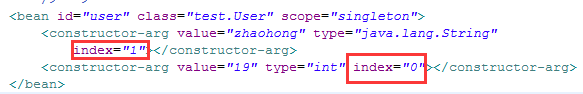
# 无参构造与有参构造

在bean标签内部，利用**constructor-arg**声明。

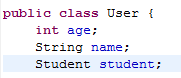
### 使用无参的构造方法

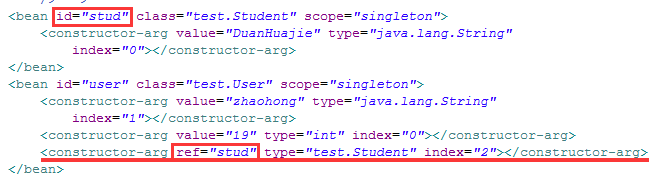


### 使用带参的构造方法

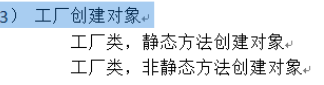


如果对象需要引入其他对象时，需要利用ref属性



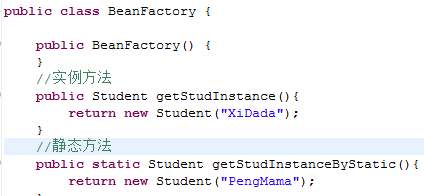


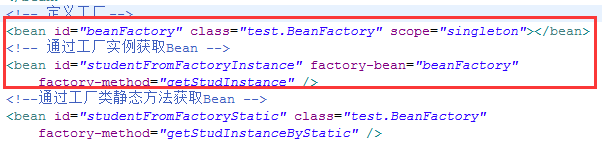
# 利用工厂创建Bean：静态方法与实例方法

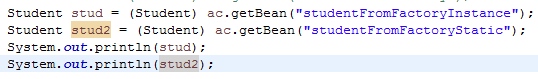


利用**工厂实例方法**，**需要利用factory-bean指定工厂对象**；首先创建工厂对象，然后创建Bean。

如果利用工厂静态方法，需要利用**class指定工厂类**。



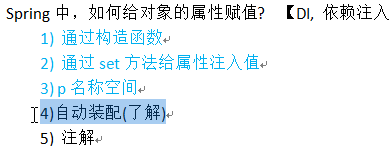






# 对象依赖关系：依赖注入

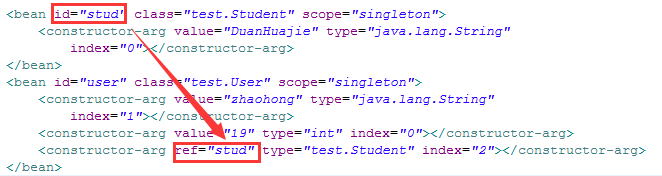
Spring中，对象属性赋值方法：



其中set方法最为常用。

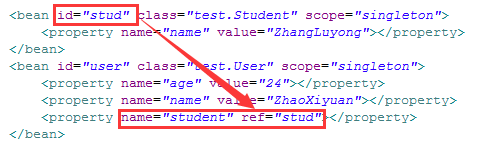
## 通过构造方法：构造注入

在bean中通过**constructor-arg**引入属性值。

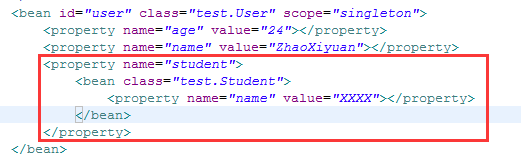


## 通过set方法给属性赋值：（常用方法）设值注入

在bean内部利用property标签，调用相应的set方法。setName，则取name。去掉set并将首字母小写即可作为property的name值。

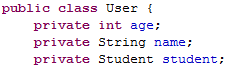


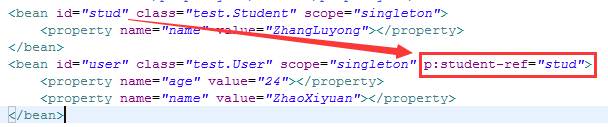
**内部Bean方法**：可以在一个Bean的内部创建所依赖的Bean。（这种方式不好维护）



## p名称空间给对象的属性注入值（Spring3.0以上）

该方法本质上也是set方法注入值，只是配置文件的写法简单些。





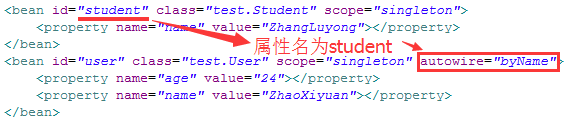
注意:**p:student与p:student-ref**的区别。前者是直接赋值，后者是引用。

## 自动装配（了解）

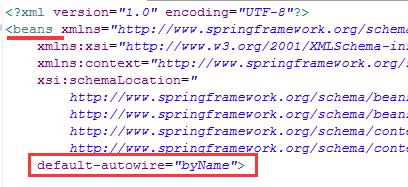
spring框架式**默认不支持自动装配**的，要想使用自动装配需要修改spring配置文件中<bean>标签的**autowire属性。**



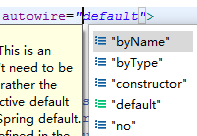




不仅对某个Bean可是设置为自动装配，还可以设置全局Bean都是自动装配。



除了byName还有bType、constructor、default、no。



Bean标签属性的autowire的五个值介绍：

### byName：根据bean的id在Spring容器中寻找bean，id名不能重复。

### byType：根据bean标签的class属性指定的类获取bean。

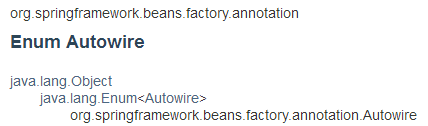


### constructor：

使用**构造方法**完成对象注入，其实也是根据**构造方法的参数类型**进行对象查找，相当于采用**byType**的方式。

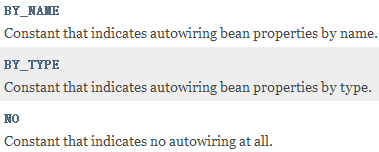
### default：表示默认采用上一级标签的自动装配的取值。如果存在多个配置文件的话，那么每一个配置文件的自动装配方式都是独立的。

### no：不支持自动装配功能。



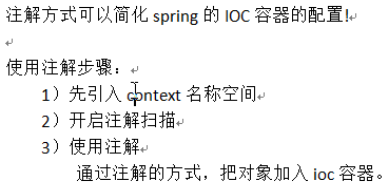
Enumeration determining autowiring status: that is, whether a bean should have its dependencies automatically injected by the Spring container using setter injection. This is a core concept in Spring DI.

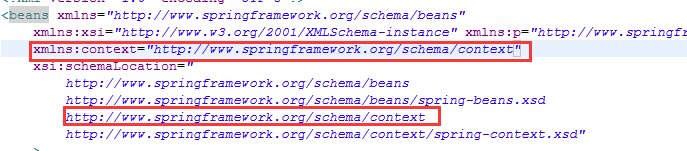
Available for use in annotation-based configurations, such as for the AspectJ AnnotationBeanConfigurer aspect.



## 注解方式

注意：一定要使**Spring与JDK版本**匹配。







# 对Bean的属性注入方法

一般说两种，也有说三种的。

1. **设值注入：通过setter方法注入；（重点）**

**能够对简单类型、引用类型、数组、List集合、Map集合赋值。**

1. **构造注入：通过构造方法注入；**

**对简单类型、引用类型赋值、数组、List集合、Map集合赋值。**

1. 接口注入：不要求掌握。

# 对Bean的各种属性赋值：采用设值注入方法

## 简单数据类型

具有对应的setter方法。



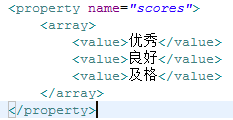
## 对象





## 数组

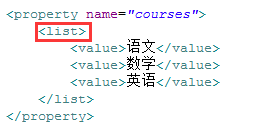




## List集合

使用list标签对List集合赋值。

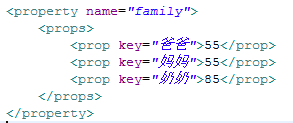




## Map集合（及Properties）

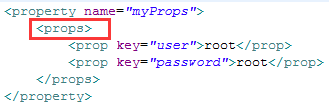
利用props标签对**Map集合**传值。





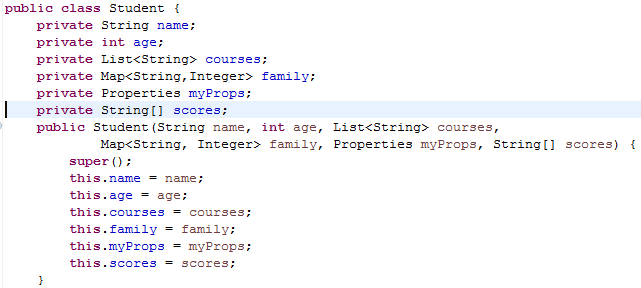
Properties实现了Map接口，也是Map集合中的一类。





# 对Bean的各种属性赋值：采用构造注入方法

注意一定要保证具有包含相应属性的构造方法，属性不可以少也不可以多。





Student [name=YYY, age=16, courses=[物理, 化学, 生物], family={妹妹=11, 姐姐=21, 哥哥=24}, myProps={height=175, weight=140}, scores=[优秀, 良好, 及格]]