## Spring框架（注解）

1. 使用注解配置Bean
2. 注解：是一个类，使用@注解名
3. 开发中，使用注解取代xml
4. 注解使用前提：添加命名空间，让spring扫描含有注解的类
5. schema命名空间：

第一步：声明命名空间（两种方式）

\* 默认：xmlns=” ”，如：xmlns="http://www.springframework.org/schema/beans"

\* 显示：xmlns:别名=” ”，如：xmlns:context="http://www.springframework.org/schema/context"

第二步：确定schema xsd文件位置

xsi:schemaLocation=”名称 位置 名称2 位置2”

1. 使用命名空间（对应上面两种方式）

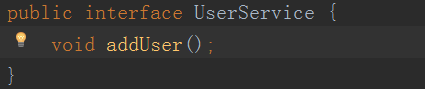
\* 默认：<标签名> 如：<bean>

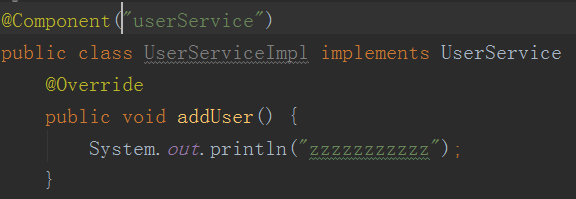
\* 显示：<别名：标签名> 如：<context:标签名>

1. @Component取代<bean class=” ”>

@Component(“id”)取代<bean id=” ” class=” ”>

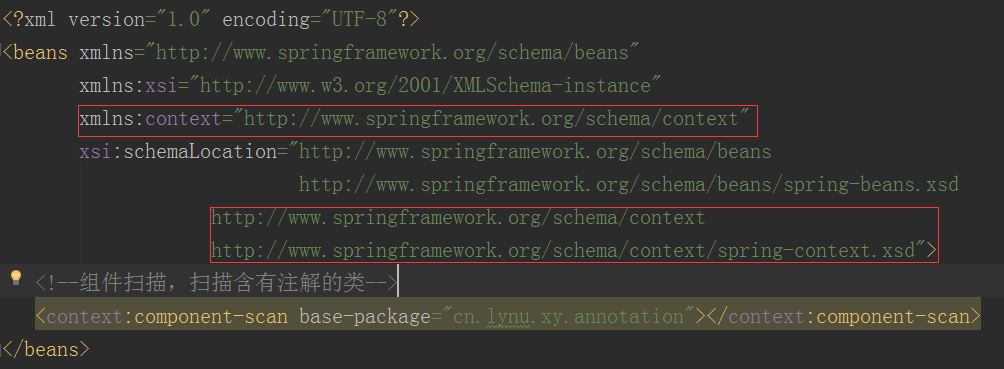
1. 例子：
2. 接口和实现类





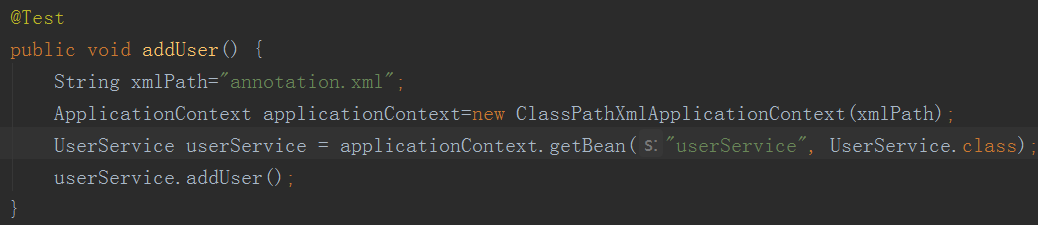
**说明：哪个类需要创建bean（即对象），就在哪个类上面添加@Component注解**

1. 配置文件



**说明：如果使用注解配置bean，需要添加上面红框命名空间**

1. 测试



1. web层使用注解
2. 使用注解创建bean

1）web层提供了3个和@Component功能一样注解，都是用来创建bean的，分别为：

\* @Repository（dao层）

\* @Service（service层）

\* @Controller（web层）

1. 使用注解设置值（依赖注入）
2. 依赖注入（就是给属性设置值），可以给私有字段设置值，也可以给setter方法设置值
3. 给普通属性注入：@Value（“具体的值”）
4. 给引用设值，两种方式

\* 按照**类型**注入：@Autowired

\* 按照**名称**注入（两种方式）：

第一种：@Autowired和@Qualifier("名称")两者一起使用

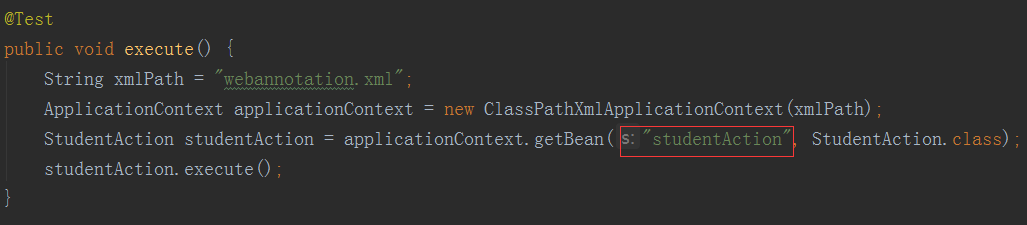
第二种：@Resource("名称")

1. 生命周期

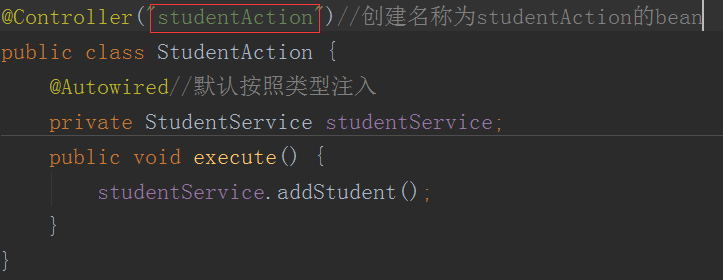
\* 初始化：@PostConstruct

\* 销毁：@PreDestroy

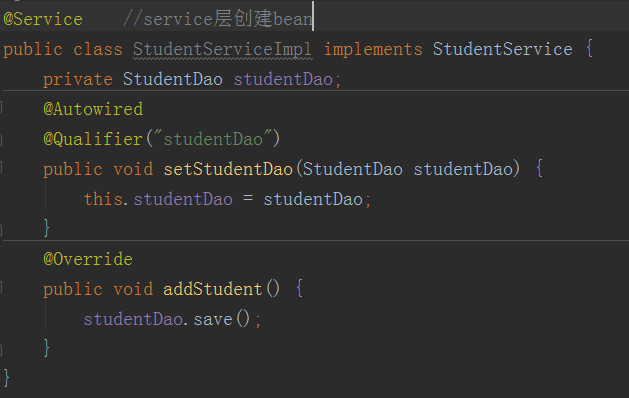
1. web层使用注解演示



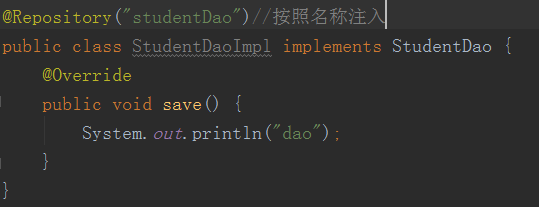
程序入口



Web层



Service层



Dao层

1. 注解和xml混合使用

