<http://blog.csdn.net/zhang854429783/article/details/6785574>

版权声明：本文为博主原创文章，未经博主允许不得转载。

很长时间没做web项目都把以前学的那点框架知识忘光了，今天把以前做的一个项目翻出来看一下发现用·@Component标记一个组件，而网上有的用@Service标记组件，我晕就查了一下资料：

[**spring**](http://lib.csdn.net/base/javaee) 2.5 中除了提供 @Component 注释外，还定义了几个拥有特殊语义的注释，它们分别是：@Repository、@Service 和 @Controller。  
在目前的 Spring 版本中，这 3 个注释和 @Component 是等效的，但是从注释类的命名上，很容易看出这 3 个注释分别和持久层、业务层和控制层（Web 层）相对应。  
虽然目前这3 个注释和 @Component 相比没有什么新意，但 Spring 将在以后的版本中为它们添加特殊的功能。  
所以，如果 Web 应用程序采用了经典的三层分层结构的话，最好在持久层、业务层和控制层分别采用上述注解对分层中的类进行注释。

@Service用于标注业务层组件

@Controller用于标注控制层组件（如struts中的action）

@Repository用于标注数据访问组件，即DAO组件

@Component泛指组件，当组件不好归类的时候，我们可以使用这个注解进行标注。

**[java]** [view plain](http://blog.csdn.net/zhang854429783/article/details/6785574) [copy](http://blog.csdn.net/zhang854429783/article/details/6785574)

1. @Service
2. **public** **class** VentorServiceImpl **implements** iVentorService {
3. }
4. @Repository
5. **public** **class** VentorDaoImpl **implements** iVentorDao {
6. }

在一个稍大的项目中，如果组件采用xml的bean定义来配置，显然会增加配置文件的体积，查找以及维护起来也不太方便。   
Spring2.5为我们引入了组件自动扫描机制，他在类路径下寻找标注了上述注解的类，并把这些类纳入进spring容器中管理。  
它的作用和在xml文件中使用bean节点配置组件时一样的。要使用自动扫描机制，我们需要打开以下配置信息：

代码

**[html]** [view plain](http://blog.csdn.net/zhang854429783/article/details/6785574) [copy](http://blog.csdn.net/zhang854429783/article/details/6785574)

1. **<?xml** version="1.0" encoding="UTF-8" **?>**
2. **<beans** xmlns="http://www.springframework.org/schema/beans"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4. xmlns:context="http://www.springframework.org/schema/context"
5. xsi:schemaLocation="http://www.springframework.org/schema/beans
6. http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
7. http://www.springframework.org/schema/context
8. http://www.springframework.org/schema/context/spring-context-2.5.xsd"**>**
10. **<context:component-scan** base-package=”com.eric.spring”**>**
11. **</beans>**

1.component-scan标签默认情况下自动扫描指定路径下的包（含所有子包），将带有@Component、@Repository、@Service、@Controller标签的类自动注册到spring容器。对标记了 Spring's @Required、@Autowired、JSR250's @PostConstruct、@PreDestroy、@Resource、JAX-WS's @WebServiceRef、EJB3's @EJB、JPA's @PersistenceContext、@PersistenceUnit等注解的类进行对应的操作使注解生效（包含了annotation-config标签的作用）。

getBean的默认名称是类名（头字母小写），如果想自定义，可以@Service(“aaaaa”)这样来指定。  
这种bean默认是“singleton”的，如果想改变，可以使用@Scope(“prototype”)来改变。

可以使用以下方式指定初始化方法和销毁方法：

**[java]** [view plain](http://blog.csdn.net/zhang854429783/article/details/6785574) [copy](http://blog.csdn.net/zhang854429783/article/details/6785574)

1. @PostConstruct
2. **public** **void** init() {
3. }
4. @PreDestroy
5. **public** **void** destory() {
6. }

注入方式：

把DAO实现类注入到action的service接口(注意不要是service的实现类)中，注入时不要new 这个注入的类，因为spring会自动注入，如果手动再new的话会出现错误，  
然后属性加上@Autowired后不需要getter()和setter()方法，Spring也会自动注入。

在接口前面标上@Autowired注释使得接口可以被容器注入，如：

**[java]** [view plain](http://blog.csdn.net/zhang854429783/article/details/6785574) [copy](http://blog.csdn.net/zhang854429783/article/details/6785574)

1. @Autowired
2. @Qualifier("chinese")
3. **private** Man man;

当接口存在两个实现类的时候必须使用@Qualifier指定注入哪个实现类，否则可以省略，只写@Autowired。

<http://crabboy.iteye.com/blog/339840>