

spring配置注解context:annotation-config和context:component-scan区别

Spring 中在使用注解（Annotation）会涉及到< context:annotation-config> 和 < context:component-scan>配置，下面就对这两个配置进行诠释。

1.context:annotation-config

< context:annotation-config> 是用于激活那些已经在spring容器里注册过的bean上面的注解，也就是显式的向Spring注册

AutowiredAnnotationBeanPostProcessor
CommonAnnotationBeanPostProcessor
PersistenceAnnotationBeanPostProcessor
RequiredAnnotationBeanPostProcessor

这四个Processor，注册这4个BeanPostProcessor的作用，就是为了你的系统能够识别相应的注解。BeanPostProcessor就是处理注解的处理器。

- 比如我们要使用@Autowired注解，那么就必须事先在 Spring 容器中声明

AutowiredAnnotationBeanPostProcessor Bean。传统声明方式如下

```
<bean class="org.springframework.beans.factory.annotation.  
AutowiredAnnotationBeanPostProcessor"/>
```

- 如果想使用@ Resource 、@ PostConstruct、@ PreDestroy等注解就必须声明

CommonAnnotationBeanPostProcessor。传统声明方式如下

```
<bean class="org.springframework.beans.factory.annotation.  
CommonAnnotationBeanPostProcessor"/>
```

- 如果想使用@PersistenceContext注解，就必须声明

PersistenceAnnotationBeanPostProcessor的Bean。

```
<bean  
class="org.springframework.beans.factory.annotation.PersistenceAnnotationBeanPostProcessor"/>
```

- 如果想使用 @Required的注解，就必须声明RequiredAnnotationBeanPostProcessor的 Bean。

同样，传统的声明方式如下：

```
<bean  
class="org.springframework.beans.factory.annotation.RequiredAnnotationBeanPostProcessor"/>
```

一般来说，像@ Resource 、@ PostConstruct、@Antowired这些注解在自动注入还是比较常用，所以如果总是需要按照传统的方式一条一条配置显得有些繁琐和没有必要，于是spring给我们提供< context:annotation-config/>的简化配置方式，自动帮你完成声明。

思考1：假如我们要使用如@Component、@Controller、@Service等这些注解，使用能否激活这些注解呢？

答案：单纯使用< context:annotation-config/>对上面这些注解无效，不能激活！

2.context:component-scan

Spring 给我提供了context:component-scan配置，如下

```
<context:component-scan base-package=" XX.XX" />
```

该配置项其实也包含了自动注入上述 四个processor 的功能，因此当使用 < context:component-scan/>后，就可以将 < context:annotation-config/> 移除了。

通过对base-package配置，就可以把controller包下 service包下 dao包下的注解全部扫描到了！

3.总结

(1) `< context:annotation-config />`：仅能够在已经在已经注册过的bean上面起作用。对于没有在spring容器中注册的bean，它并不能执行任何操作。

(2) `< context:component-scan base-package="XX.XX"/>`：除了具有上面的功能之外，还具有自动将带有@Component, @service, @Repository等注解的对象注册到spring容器中的功能。

思考2：如果同时使用这两个配置会不会出现重复注入的情况呢？

答案：`< context:annotation-config />`和 `< context:component-scan>` 同时注册并不是前面的会被忽略，而是看哪个声明在前，解析的时候如果当前还没有注册这些PostProcessor，就会新建一个BeanDefinition然后注册。另外spring3.0还添加了ConfigurationClassPostProcessor用于@Configuration注解。

`< mvc:annotation-driven/>`

`< mvc:annotation-driven/>`从 标签的shecma就能看出来，mvc主要就是为了Spring MVC来用的，提供Controller请求转发，json自动转换等功能。相比上面的两个shecma是context开头，那么主要是解决spring容器的一些注解。

`< mvc:annotation-driven />` 是一种简写形式，完全可以手动配置替代这种简写形式，简写形式可以让初学都快速应用默认配置方案。会自动注册DefaultAnnotationHandlerMapping与AnnotationMethodHandlerAdapter 两个bean，是spring MVC为@Controller分发请求所必须的。并提供了：数据绑定支持，@NumberFormatannotation支持，@DateTimeFormat支持，@Valid支持，读写XML的支持（JAXB），读写JSON的支持（Jackson）。

在实际开发使用SpringMVC开启这个配置，否则会出现一些功能不能正常使用！