spring学习笔记—基于注解的配置方式

|  |  |
| --- | --- |
|  | 🢂 内容概览 |
|  | Why：此文档用来做什么？它存在的意义是什么？为解决什么问题？   |  | | --- | |  |   What：当前包含了那些内容？   |  | | --- | |  |   How：此文档应如何参考？   |  | | --- | |  |   Who：此文档适用于那些人员阅读参考？   |  | | --- | |  | |

目录

[1 基于注解的容器配置 3](#_Toc478750170)

[1.1 @Required：RequiredAnnotationBeanPostProcessor 3](#_Toc478750171)

[1.2 @Autowired：AutowiredAnnotationBeanPostProcessor 3](#_Toc478750172)

[1.3 @Primary 6](#_Toc478750173)

[1.4 @Qualifier 7](#_Toc478750174)

[1.5 @Resource与@Autowired的区别： 8](#_Toc478750175)

[1.5.1 定制@Qualifier注解 8](#_Toc478750176)

[1.5.1.1 有参注解 8](#_Toc478750177)

[1.5.1.2 无参注解 9](#_Toc478750178)

[1.5.1.3 多参注解 10](#_Toc478750179)

[1.5.2 CustomAutowireConfigurer 11](#_Toc478750180)

[1.6 @Resource：CommonAnnotationBeanPostProcessor 12](#_Toc478750181)

[1.7 @PostConstruct与@PreDestory：CommonAnnotationBeanPostProcessor 12](#_Toc478750182)

# 基于注解的容器配置

1. 通过RequiredAnnotationBeanPostProcessor，AutowiredAnnotationBeanPostProcessor，CommonAnnotationBeanPostProcessor，PersistenceAnnotationBeanPostProcessor ，spring能够支持注解配置方式；
2. 注解注入要先于xml配置的注入，故后配置的内容会覆盖先前的配置；
3. 支持注解的配置方式如下：

|  |
| --- |
|  |

1. 注意：annotation-config只能够注入相同容器中的bean，如果使用WebApplicationContext，则只会识别Controller；不会注入service；

## @Required：RequiredAnnotationBeanPostProcessor

标识一个必需注入的bean，通过setter注入对应类型的bean；

|  |
| --- |
|  |

## @Autowired：AutowiredAnnotationBeanPostProcessor

1. 通过setter注入bean：

|  |
| --- |
|  |

1. 用在在任意方法上，支持多参数；

|  |
| --- |
|  |

1. 用在构造器或者属性上；

|  |
| --- |
|  |

1. 用在数组或者集合上，将容器中所有相同类型的bean都注入；

|  |
| --- |
|  |

Set集合

|  |
| --- |
|  |

注：通过实现org.springframework.core.Ordered接口或者使用@Ordered,@Priority注解来设置bean在集合中的顺序；

1. 可以用来注入map，key即为map的名称，value即为容器中所有指定类型的bean;

|  |
| --- |
|  |

1. 默认情况下，注入都是必需的，可以指定注入是非必需的；对于构造器而言，每个类只能有一个构造器注入指定是required，但可以有多个非required的构造器注入，此时，spring会使用贪心策略，满足尽可能多个的参数的构造器；

|  |
| --- |
|  |

1. 可以注入容器相关的接口以及子接口，bean，比如：BeanFactory，ApplicationContext，Envrionment，ResourceLoader，ApplicationEventPublisher，MessageSource，以及子接口ConfigurableApplicationConext，ResourcePatternResover等；

|  |
| --- |
|  |

注：@Autowried，@Inject，@Resource，@Value这些注解都是通过spring的BeanPostProcessor来处理的，故不能够在BeanPostProcessor和BeanFactoryPostProcessor中使用；必须使用xml或者@Bean注解进行注入；

1. 支持泛型自动注入，不同的泛型注入不同的bean；

|  |
| --- |
|  |

## @Primary

使用@Autowried注解时，经常遇到注入bean有多个候选者的情况，通过@Primary可以指定优先被注入的候选Bean；

|  |
| --- |
|  |

## @Qualifier

当@Autowired存在多个候选bean时，用于精确筛选要注入的Bean；

|  |
| --- |
|  |

也可以用在构造器参数中：

|  |
| --- |
|  |

对应的xml配置如下：

|  |
| --- |
|  |

## @Resource与@Autowired的区别：

1. @Autowired默认使用By-type方式，如果需要精确匹配具体的Bean，需要使用@Qualifier配合；@Resource默认使用By-name方式；
2. @Autowired可用于属性、构造器、多参函数，支持使用@Qualifier对参数筛选；@Resource仅可以用于属性、仅有一个参数的setter方法；

### 定制@Qualifier注解

#### 有参注解

1. 自定义注解

|  |
| --- |
|  |

1. 使用注解

|  |
| --- |
|  |

1. 配置xml

|  |
| --- |
|  |

#### 无参注解

1. 定义注解

|  |
| --- |
|  |

1. 使用注解

|  |
| --- |
|  |

1. 配置xml；

|  |
| --- |
|  |

#### 多参注解

1. 定义注解

|  |
| --- |
|  |

1. 使用注解

|  |
| --- |
|  |

1. 配置xml

|  |
| --- |
|  |

### CustomAutowireConfigurer

一种BeanFactoryPostProcessor的实现，允许定制自己的Qualifier注解，即使没有被@Qualifier注解标注；如下为注册注解示例：

|  |
| --- |
|  |

AutowireCandidateResolver通过如下策略筛选候选bean：

1. Bean定义中的autowire-candidate
2. <beans/>中定义的default-autowire-candidates  模式字符串
3. @Qualifier注解以及任何使用CustomAutowireConfigurer注册的注解；

## @Resource：CommonAnnotationBeanPostProcessor

1. 默认使用By-name方式注入bean；如果未显式指定name属性，则根据属性名称或者setter方法名自动解析bean的名称；

|  |
| --- |
|  |

|  |
| --- |
|  |

1. 可以注入容器相关的接口以及子接口，bean，比如：BeanFactory，ApplicationContext，Envrionment，ResourceLoader，ApplicationEventPublisher，MessageSource，以及子接口ConfigurableApplicationConext，ResourcePatternResover等；容器会先根据name查找有无相关bean，如果没有，则去查找容器相关的基础类型bean；

|  |
| --- |
|  |

## @PostConstruct与@PreDestory：CommonAnnotationBeanPostProcessor

CommonAnnotationBeanPostProcessor不仅处理@Resource注解，也处理生命周期相关的注解；

|  |
| --- |
|  |