Atitit spring重要注解

目录

[1.1. 注解历史 1](#_Toc8096)

[1.2. Spring帮助我们管理Bean分为两个部分，一个是注册Bean，一个装配Bean。  1](#_Toc12697)

[1.3. @Bean 创建来自第三方库的组件 1](#_Toc20469)

[1.4. 3、@Resource和@Autowired 2](#_Toc1466)

[1.5. 8、@Component 3](#_Toc18640)

[1.6. 9、@Repository 3](#_Toc19854)

## 注解历史

众所周知，@Component注解是在Spring 2.5版本引入的，以便于可以通过路径扫描的方式来替换配置文件。  
@Bean是在Spring3.0版本引入的，可以配合使用@Configuration注解来达到完全替换配置文件的目的。

## Spring帮助我们管理Bean分为两个部分，一个是注册Bean，一个装配Bean。

完成这两个动作有三种方式，一种是使用自动配置的方式、一种是使用JavaConfig的方式，一种就是使用XML配置的方式。

## @Bean 创建来自第三方库的组件

@Bean 需要在配置类中使用，即类上需要加上@Configuration注解

@Component（和@Service和@Repository）用于自动检测和使用类路径扫描自动配置bean。注释类和bean之间存在隐式的一对一映射（即每个类一个bean）。这种方法对需要进行逻辑处理的控制非常有限，因为它纯粹是声明性的。  
@Bean用于显式声明单个bean，而不是让Spring像上面那样自动执行它。它将bean的声明与类定义分离，并允许您精确地创建和配置bean。  
那我们在什么时候会使用@Bean？让我们假设下，如果你想要创建来自第三方库的组件（您没有源代码，因此您无法使用@Component注释其类），因此无法使用@Component进行自动配置，这种时候使用@Bean则可以毫不费力的完成。  
又比如，具体采用哪个实现是依赖于某些参数，在这种情况下@Bean无疑是最合适的选择。

## **3、@Resource和@Autowired**

@Resource和@Autowired都是做bean的注入时使用，其实@Resource并不是Spring的注解，它的包是javax.annotation.Resource，需要导入，但是Spring支持该注解的注入。

@Autowired注解是按照类型（byType）装配依赖对象，默认情况下它要求依赖对象必须存在，如果允许null值，可以设置它的required属性为false。如果我们想使用按照名称（byName）来装配，可以结合@Qualifier注解一起使用。如下：

（2）@Resource

@Resource默认按照ByName自动注入，由J2EE提供，需要导入包javax.annotation.Resource。@Resource有两个重要的属性：name和type，而Spring将@Resource注解的name属性解析为bean的名字，而type属性则解析为bean的类型。所以，如果使用name属性，则使用byName的自动注入策略，而使用type属性时则使用byType自动注入策略。如果既不制定name也不制定type属性，这时将通过反射机制使用byName自动注入策略。

@Resource装配顺序：

①如果同时指定了name和type，则从Spring上下文中找到唯一匹配的bean进行装配，找不到则抛出异常。

②如果指定了name，则从上下文中查找名称（id）匹配的bean进行装配，找不到则抛出异常。

③如果指定了type，则从上下文中找到类似匹配的唯一bean进行装配，找不到或是找到多个，都会抛出异常。

④如果既没有指定name，又没有指定type，则自动按照byName方式进行装配；如果没有匹配，则回退为一个原始类型进行匹配，如果匹配则自动装配。

@Resource的作用相当于@Autowired，只不过@Autowired按照byType自动注入。

## **8、**@Component****

相当于通用的注解，当不知道一些类归到哪个层时使用，但是不建议。

## **9、@Repository**

用于注解dao层，在daoImpl类上面注解。

@Component 和 @Bean 的区别 - LKMMKL - CSDN博客