# 第二章 装配Bean

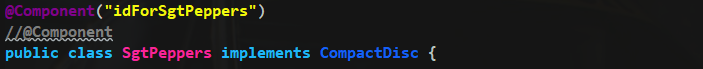
Spring配置可选方案优先级（建议）

1. 尽可能的使用自动配置机制，显式配置越少越好；
2. 使用类型安全并且比XML更强大的javaConfig；
3. 最后，当想使用便利的XML命名空间，并且在XML中没有同意配置时，才应该使用XML。

## @Component – 组件

在类上使用Component注解，以表明该类作为组件类，并通过XML或javaConfig告知Spring为这个类创建Bean。

Spring上下文会自动给所有的Bean一个id，默认的id为组件类类名（首字母小写），也可以指定。



如不指定时默认id为：sgtPeppers。

另一种为Bean命名的方式是使用java依赖注入规范提供的@Named注解，可达到同样效果，但不推荐使用。

## @ComponentScan – 启用组件扫描

用ComponentScan注解启用组件扫描，查找带有Component注解的类，从而为其创建Bean。

ComponentScan注解默认扫描与配置类相同的包。

1. 使用basePackages指定要扫描的包（字符串数组的形式）
2. 或用basePackageClasses指定要扫描的组件所在的包（通过给出一个目标包中的组件类的Class），采用这种形式时可在目标包中创建一个空标记接口以避免引用任何实际的应用程序代码。



使用XML启用组件扫描：



## @Autowire – 自动装配

Autowire注解可注解于构造器、类成员变量、方法。不管是构造器、Setter方法还是其他的方法，Spring都会尝试装配满足方法参数上所声明的依赖。

如果没有匹配的依赖，在上下文创建时Spring会抛出一个异常，通过将Autowire的required成员设置为false可避免该异常，但此时因考虑目标Bean在后续使用时可能遇到的NullPointerException。

如果有多个Bean满足依赖关系的话，Spring也会抛出异常。

Java依赖注入规范提供了@Inject注解，可达到与Autowire一样的效果。

