

【第29话：不会还是在说Spring进行new对象就是IoC吧】

Hello 小伙伴们，这节课给大家依然给大家带来一个高频面试题：“请说一下IoC/DI”。

这道面试题已经被问了很多年了。

我相信很多在看视频的小伙伴里面被问到IoC/DI时会这样回答：

- IoC 英文全称 Inversion of Control 中文名称控制反转。就是本来应该由程序员new对象，现在交给了Spring框架去完成就是控制反转。
- DI 英文全称Dependency Injection 中文名称依赖注入。就是一个类的属性（方法参数）是另一个类，在Spring容器内部可以把另一个类的Bean赋值给这个类的属性或参数。

那小伙伴们，我在问大家一个问题：“为什么说IoC/DI”是同一件事情？

有的小伙伴会说因为在实例化Bean时，我们可以给Bean进行注入属性值，发生IoC时可以发生DI所以认为IoC和DI是同一件事情。有意思的是还会举证，说在哪里的博客看过解释，在哪个视频看过这样的解释。

但是不管谁的解释，都没有官方的解释正确。我们一起看下Spring官方文档中对IoC/DI的解释。

先看看IoC的官方解释：

网址：<https://docs.spring.io/spring-framework/docs/5.3.24/reference/html/core.html#spring-core>

关键部分截图：

1.1. Introduction to the Spring IoC Container and Beans

This chapter covers the Spring Framework implementation of the Inversion of Control (IoC) principle. IoC is also known as dependency injection (DI). It is a process whereby objects define their dependencies (that is, the other objects they work with) only through constructor arguments, arguments to a factory method, or properties that are set on the object instance after it is constructed or returned from a factory method. The container then injects those dependencies when it creates the bean. This process is fundamentally the inverse (hence the name, Inversion of Control) of the bean itself controlling the instantiation or location of its dependencies by using direct construction of classes or a mechanism such as the Service Locator pattern.

在来看看DI的官方解释：

1.4.1. Dependency Injection

Dependency injection (DI) is a process whereby objects define their dependencies (that is, the other objects with which they work) only through constructor arguments, arguments to a factory method, or properties that are set on the object instance after it is constructed or returned from a factory method. The container then injects those dependencies when it creates the bean. This process is fundamentally the inverse (hence the name, Inversion of Control) of the bean itself controlling the instantiation or location of its dependencies on its own by using direct construction of classes or the Service Locator pattern.

上面这两个解释中有一句非常重要的话：“is a process” 翻译过来：“是一个过程”。敲黑板了，这就是重点。很多人认为的IoC就是Spring 帮助new对象这点，DI就是注入属性值这点。错就错在了我们认为的点，在Spring官方解释中应该是一个面，一个完整的过程。

Spring IoC/DI官方回答版

所以Spring IoC/DI最正确的解释是：IoC 控制反转，又称DI依赖注入。它代表的是一个过程(Process)，整个过程包含了Spring 在容器中创建Bean，并从容器中获取到其他Bean注入到当前Bean。在实例化Bean时可以使用构造方法，也可以使用工厂。如果想要给Bean注入属性值可以使用有参构造方法，也可以使用属性的setter方法进行注入。但无论是实例化还是设置属性值，这些事情都是反生在Spring容器内部的。

上面的回答不仅仅包含了IoC/DI的概念，还包含了Spring中另外两个常见的面试题：“Spring中实例化Bean的几种方式”和“Spring中注入的几种方式”。

如果就按照上面的解释，在实际与面试官交谈时可能只有几十秒就回答完了，如果想让自己的会答更加丰富可以在多说一些：

- IoC作为Spring的几大功能之一，也是Spring其他基础的基础。没有Spring的IoC容器，也没有Spring的AOP。没有Spring的AOP也不会有Spring的声明式事务。所以Spring IoC是整个Spring框架的基础。
- 无论是Spring还是目前的Spring Boot框架。在项目中可以很好的把其他框架或技术整合到一起，也多亏了Spring IoC。在企业项目中，每个项目都可能有几十或几百个类需要管理，这些类都交给了Spring IoC容器管理后，类之间相互注入变得更加方便，更加容易。这点只要用过Spring或Spring Boot做项目的人都会有很深的感触。