

16_源码剖析：Spring IOC 容器 Bean 初始化流程？

1、开篇

上节课介绍了 Spring IOC 查找 Bean 出现的异常，这些异常都是 `BeansException` 的子接口。其中 `NoSuchBeanDefinitionException`（Bean 不存在）和 `NoUniqueBeanDefinitionException`（Bean 不唯一）两种异常的使用场景是最为常见的，我们通过测试代码的方式将两种场景的异常做了介绍。

通过 Spring IoC 容器初始化 Bean 的流程，在之前的章节中介绍了 Spring IoC 容器是如何从配置文件中读取 Bean 信息，并且将其转换为 `BeanDefinition` 并且注册到 Spring IoC 容器中进行统一管理的。今天讲的是，在 Spring IoC 容器中的 Bean 实例是如何被创建出来的，也就是解决如何从 Spring IoC 取出 Bean 进行使用。今天有如下内容：

- 创建 Bean 的时序图

2、创建 Bean 的时序图

由于一个 Bean 的创建经历了若干个类，调用了很多方法，为了将这个过程理解清楚，这里根据图 1 的时序图给大家展开描述。这里涉及到几个主要调用的类，我们会根据重点，按照图上的红色数字标号进行讲解。

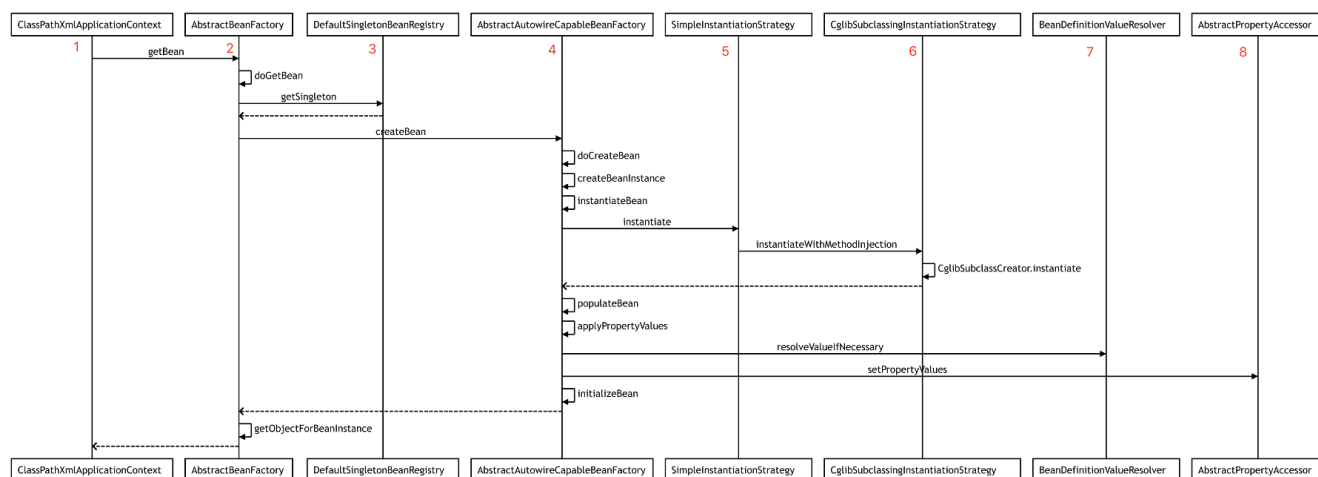


图 1 创建 Bean 的时序图

如图 1 所示，按照红色标号的顺序如下：

1. 首先通过 `ClassPathXmlApplicationContext` 获取要创建的 `Bean`，这里的方法是 `getBean`。
2. 在 `AbstractBeanFactory` 中会通过调用 `doGetBean` 方法完成获取 `Bean` 的操作。
3. 此时 `DefaultSingletonBeanResigstry` 会执行 `getSingleton` 通过单例的方式返回 `Bean`。
4. 在获取 `Bean` 以后会通过 `createBean` 方法调用 `AbstractAutowireCapableBeanFactory` 中的方法对 `Bean` 进行初始化。这里重要的三个方式分别是：`doCreateBean`、`createBeanInstance`、`instantiateBean`。从图上可以看出 `Bean` 的初始化主要过程都是在这个类中完成的。
5. 紧接着通过 `instantiate` 调用 `SimpleInstantiationStrategy` 中的 `instantiateWithMethodInjection` 做初始化的操作。
6. `CglibSubclassingInstantiationStrategy` 作为 `CGLIB` 的代理实现策略类，通过 `CglibSubclassCreator.instantiate` 方法进行初始化，并且返回 `AbstractAutowireCapableBeanFactory`。此时 `AbstractAutowireCapableBeanFactory` 会接着执行 `populateBean` 和 `applyPropertyValues` 方法对 `Bean` 进行填充，以及设置属性值。
7. 期间会通过调用 `BeanDefinitionValueResolver` 中的 `resolveValueIfNecessary` 方法完成对 `BeanDefinition` 中的值填充。
8. 以及调用 `AbstractPropertyAccessor` 中的 `setPropertyValues` 方法进行属性值设置。完成上述操作以后，在 `AbstractAutowireCapableBeanFactory` 中执行 `initializeBean` 方法返回给 `AbstractBeanFactory`，然后由 `AbstractBeanFactory` 的 `getObjectForBeanInstance` 方法返回 `Bean` 给调用的客户端，完成 `Bean` 的初始化过程。

3、总结

本节课介绍了 Spring IOC 容器初始化 `Bean` 的全过程，涉及到 `ClassPathXmlApplicationContext`、`AbstractBeanFactory`、`DefaultSingletonBeanResigstry`、`AbstractAutowireCapableBeanFactory`、

SimpleInstantiationStrategy、CglibSubclassingInstantiationStrategy、
BeanDefinitionValueResolver 8 个类以及对应的方法。

下节课，会围绕 Spring IOC 依赖注入方式给大家做讲解，下期见，拜拜。