

Spring不仅提供了一个进行快速开发的基础框架，而且还提供了很多可扩展的接口，用于满足一些额外的开发需求，本篇将对常用的可扩展接口进行归纳总结。

InitializingBean接口

InitializingBean接口中只有一个afterPropertiesSet方法，从方法的名称上很容易理解，这个方法是在Bean的属性都设置值后被调用，用于完成一些初始化工作。当然，在Spring的配置文件中init-method的配置也是在Bean的属性都设置值后被调用，用于完成一些初始化工作，不过在执行顺序上，接口的方法先于配置。值得注意的是，这两种方式都是用于完成一些初始化工作，所以相应的方法中不要编写一些复杂且执行时间很长的逻辑。

DisposableBean接口

DisposableBean接口中只有一个destroy方法，该方法会在Bean被销毁、生命周期结束之前被调用，用于做一些销毁的收尾工作。同样，在Spring的配置文件中destroy-method配置也完成同样的工作，不过在执行顺序上，接口的方法先于配置。

ApplicationContextAware接口

ApplicationContextAware 中 只 有 一 个 setApplicationContext 方法 。 实 现 了 ApplicationContextAware接口的类，可以在该Bean被加载的过程中获取Spring的应用上下文ApplicationContext，通过ApplicationContext可以获取Spring容器内的很多信息。

BeanFactoryAware接口

BeanFactoryAware接口中只有一个setBeanFactory方法。实现了BeanFactoryAware接口的类，可以在该Bean被加载的过程中获取加载该Bean的BeanFactory，同时也可以获取这个BeanFactory中加载的其它Bean。

FactoryBean接口

FactoryBean接口可以实现Bean实例化的个性定制，让Spring容器加载我们想要的Bean。实现了FactoryBean接口的类，可以通过实现getObject方法，实现加载我们想要的Bean。

BeanPostProcessor接口

BeanPostProcessor 接口中有两个方法，分别为postProcessBeforeInitialization和postProcessAfterInitialization。实现了BeanPostProcessor接口的类，会在每个Bean初始化(即调用setter)之前和之后，分别调用这个类中的postProcessBeforeInitialization方法和postProcessAfterInitialization方法，实现初始化的逻辑控制。

InstantiationAwareBeanPostProcessor接口

InstantiationAwareBeanPostProcessor 接 口 中 ， 常 用 的 方 法 是 postProcessBeforeInstantiation和postProcessAfterInstantiation。每个Bean的实例化(即调用构造函数)之前和之后，会分别调用实现了该接口的类中的postProcessBeforeInstantiation和postProcessAfterInstantiation方法。

BeanFactoryPostProcessor接口

BeanFactoryPostProcessor接口中只有postProcessBeanFactory方法。实现了该接口的类，可以在Bean被创建之前，获取容器中Bean的定义信息，并且可以进行修改。实现类中的postProcessBeanFactory方法只会被执行一次，且先于BeanPostProcessor接口的方法。