

spring内部提供了很多表示spring容器的接口和对象，我们来看看比较常见的几个容器接口和具体的实现类。

BeanFactory接口

`org.springframework.beans.factory.BeanFactory`

spring容器中具有代表性的容器就是BeanFactory接口，这个是spring容器的顶层接口，提供了容器最基本的功能。

常用的几个方法

//按bean的id或者别名查找容器中的bean

Object `getBean(String name)` throws BeansException

//这个是一个泛型方法，按照bean的id或者别名查找指定类型的bean，返回指定类型的bean对象

<T> T `getBean(String name, Class<T> requiredType)` throws BeansException;

//返回容器中指定类型的bean对象

<T> T `getBean(Class<T> requiredType)` throws BeansException;

//获取指定类型bean对象的获取器

<T> ObjectProvider<T> `getBeanProvider(Class<T> requiredType);`

ApplicationContext接口

`org.springframework.context.ApplicationContext`

这个接口继承了BeanFactory接口，所以内部包含了BeanFactory所有的功能，并且在其上进行了扩展，增加了很多企业级功能，比如AOP、国际化、事件支持等等。

ClassPathXmlApplicationContext类

`org.springframework.context.support.ClassPathXmlApplicationContext`

这个类实现了ApplicationContext接口，注意一下这个类名称包含了ClassPath Xml，说明这个容器类可以从classpath中加载bean xml配置文件，然后创建xml中配置的bean对象。

AnnotationConfigApplicationContext类

`org.springframework.context.annotation.AnnotationConfigApplicationContext`

这个类也实现了ApplicationContext接口，注意其类名包含了Annotation和config两个单词，上面我们有说过，bean的定义支持xml的方式和注解的方式，当我们使用注解的方式定义bean的时候，就需要用到这个容器来装载了，这个容器内部会解析注解来构建和管理需要的bean。

注解的方式相对于xml方式更方便一些，也是我们比较推荐的方式

