<https://www.cnblogs.com/yjbjingcha/p/6752265.html>

## [Spring在代码中获取bean的几种方式](https://www.cnblogs.com/yjbjingcha/p/6752265.html)

方法一：在初始化时保存ApplicationContext对象   
方法二：通过Spring提供的utils类获取ApplicationContext对象   
方法三：继承自抽象类ApplicationObjectSupport   
方法四：继承自抽象类WebApplicationObjectSupport   
方法五：实现接口ApplicationContextAware   
方法六：通过Spring提供的ContextLoader

**获取spring中bean的方式总结：**

##### 方法一：在初始化时保存ApplicationContext对象

1. ApplicationContext ac = **new** FileSystemXmlApplicationContext("applicationContext.xml");
2. ac.getBean("userService");//比如：<bean id="userService" class="com.cloud.service.impl.UserServiceImpl"></bean>

说明：这样的方式适用于採用Spring框架的独立应用程序，须要程序通过配置文件手工初始化Spring的情况。

##### 方法二：通过Spring提供的工具类获取ApplicationContext对象

1. ApplicationContext ac1 = WebApplicationContextUtils.getRequiredWebApplicationContext(ServletContext sc);
2. ApplicationContext ac2 = WebApplicationContextUtils.getWebApplicationContext(ServletContext sc);
3. ac1.getBean("beanId");
4. ac2.getBean("beanId");

说明：这样的方式适合于採用Spring框架的B/S系统，通过ServletContext对象获取ApplicationContext对象。然后在通过它获取须要的类实例。上面两个工具方式的差别是，前者在获取失败时抛出异常。后者返回null。

##### 方法三：继承自抽象类ApplicationObjectSupport

说明：抽象类ApplicationObjectSupport提供getApplicationContext()方法。能够方便的获取ApplicationContext。

Spring初始化时。会通过该抽象类的setApplicationContext(ApplicationContext context)方法将ApplicationContext 对象注入。

##### 方法四：继承自抽象类WebApplicationObjectSupport

说明：类似上面方法。调用getWebApplicationContext()获取WebApplicationContext

##### 方法五：实现接口ApplicationContextAware

说明：实现该接口的setApplicationContext(ApplicationContext context)方法，并保存ApplicationContext 对象。Spring初始化时，会通过该方法将ApplicationContext对象注入。

下面是实现ApplicationContextAware接口方式的代码，前面两种方法类似：

1. **public** **class** SpringContextUtil **implements** ApplicationContextAware {
3. // Spring应用上下文环境
4. **private** **static** ApplicationContext applicationContext;
6. /\*\*
7. \* 实现ApplicationContextAware接口的回调方法。设置上下文环境
8. \*
9. \* @param applicationContext
10. \*/
11. **public** **void** setApplicationContext(ApplicationContext applicationContext) {
12. SpringContextUtil.applicationContext = applicationContext;
13. }
15. /\*\*
16. \* @return ApplicationContext
17. \*/
18. **public** **static** ApplicationContext getApplicationContext() {
19. **return** applicationContext;
20. }
22. /\*\*
23. \* 获取对象
24. \*
25. \* @param name
26. \* @return Object
27. \* @throws BeansException
28. \*/
29. **public** **static** Object getBean(String name) **throws** BeansException {
30. **return** applicationContext.getBean(name);
31. }
32. }

尽管，spring提供的后三种方法能够实如今普通的类中继承或实现对应的类或接口来获取spring 的ApplicationContext对象，可是在使用是一定要注意实现了这些类或接口的普通java类一定要在Spring 的配置文件applicationContext.xml文件里进行配置。否则获取的ApplicationContext对象将为null。

##### 方法六：通过Spring提供的ContextLoader

1. WebApplicationContext wac = ContextLoader.getCurrentWebApplicationContext();
2. wac.getBean(beanID);

最后提供一种不依赖于servlet,不须要注入的方式。可是须要注意一点，在server启动时。Spring容器初始化时，不能通过下面方法获取Spring 容器，细节能够查看spring源代码org.springframework.web.context.ContextLoader。

## [#1楼](https://www.cnblogs.com/yjbjingcha/p/6752265.html#3723361)

不错

[支持(0)反对(0)](javascript:void(0);)

2017-06-27 10:16 | [西风瘦马justin](https://www.cnblogs.com/justinwxt/) 

## [#2楼](https://www.cnblogs.com/yjbjingcha/p/6752265.html#3972524)

感谢 很详细

[支持(0)反对(0)](javascript:void(0);)

2018-05-11 20:05 | [Master.L](https://www.cnblogs.com/Mster-L/) 