[**Mr.Johness**](http://www.cnblogs.com/Johness/)

阿何的程序人生

[Web项目中获取SpringBean——在非Spring组件中获取SpringBean](http://www.cnblogs.com/Johness/archive/2012/12/25/2833010.html)

　　最近在做项目的时候我发现一个问题：Spring的IOC容器不能在Web中被引用(或者说不能被任意地引用)。我们在配置文件中让Spring自动装配，但并没有留住ApplicationContext的实例。我们如果希望在我们的项目中任何位置都能拿到同一个ApplicationContext来获取IOC容器中的资源，就要让Spring将上下文环境填充到我们能获取的地方，比如下面的做法(来自网络资源)：

　　自定义一个工具类，实现自ApplicationContextAware接口，接口的方法是setApplicationContext，我们实现它，并让其为我们服务，因为Spring在load自己的时候会将上下文环境填充进来。我们所要做的就是将得到的ApplicationContext保存下来用。



1 package org.coderecord.ccms.web.action.util;

2

3 import org.springframework.beans.BeansException;

4 import org.springframework.context.ApplicationContext;

5 import org.springframework.context.ApplicationContextAware;

6

7 /\*\*

8 \* Spring IOC上下文工具类

9 \*

10 \* @author Ryan Shaw

11 \*

12 \*/

13 public class SpringUtil implements ApplicationContextAware {

14

15 /\*\*

16 \* 当前IOC

17 \*/

18 private static ApplicationContext applicationContext;

19

20 /\*\*

21 \* 设置当前上下文环境，此方法由spring自动装配

22 \*/

23 @Override

24 public void setApplicationContext(ApplicationContext arg0)

25 throws BeansException {

26 applicationContext = arg0;

27 }

28

29 /\*\*

30 \* 从当前IOC获取bean

31 \*

32 \* @param id

33 \* bean的id

34 \* @return

35 \*/

36 public static Object getObject(String id) {

37 Object object = null;

38 object = applicationContext.getBean(id);

39 return object;

40 }

41

42 }



　　上文的类就是我们要用的，而其中的setApplicationContext是接口中需要实现的，Spring会自动进行填充。我们在Spring的配置文件中注册一下：

1 <bean id="springUtil" class="org.coderecord.ccms.web.action.util.SpringUtil" />

　　这样就可以了，Spring把我们需要的东西给我们了。

　　我们就可以在需要的地方这样做：

1 YouClass obj = (YouClass)SpringUtil.getObject("beanid");

　　当然，前提是你需要让Spring自动装配哦。

　　以下为方法二，使用了注解和静态化的方式来产生SpringFactory对象

　　来自网络资源，修改于2013-04-11 15:25:57

　　上文的方法有个麻烦的地方：需要配置。而Spring2.5及之后的版本实际上加入了注解的方式进行依赖项的注入，使用如下代码也许更好：



1 package org.ahe.util;

2

3 import org.springframework.beans.factory.BeanFactory;

4 import org.springframework.beans.factory.annotation.Autowired;

5 import org.springframework.web.context.support.SpringBeanAutowiringSupport;

6

7 public class SpringWiredBean extends SpringBeanAutowiringSupport {

8

9 /\*\*

10 \* 自动装配注解会让Spring通过类型匹配为beanFactory注入示例

11 \*/

12 @Autowired

13 private BeanFactory beanFactory;

14

15 private SpringWiredBean() {

16 }

17

18 private static SpringWiredBean instance;

19

20 static {

21 // 静态块，初始化实例

22 instance = new SpringWiredBean();

23 }

24

25 /\*\*

26 \* 实例方法

27 \* 使用的时候先通过getInstance方法获取实例

28 \*

29 \* @param beanId

30 \* @return

31 \*/

32 public Object getBeanById(String beanId) {

33 return beanFactory.getBean(beanId);

34 }

35

36 public static SpringWiredBean getInstance() {

37 return instance;

38 }

39 }



　　如果使用@Autowired注解自动装配的话，继承SpringBeanAutowiringSupport类是不能少的。当然，使用@Component等注解也是可以的。使用注解的话配置就需要改动了，不过因为我们为支持Spring注解的配置是可以多用的，所以还好。如下：

1 <context:component-scan base-package="org.ahe"></context:component-scan>

　　该配置即可让org.ahe下所有包(您依然可以通过子标签的正则表达式匹配来进行更多设置)下的注解起作用。