[**java中借助Spring获取所有带有指定注解的接口、类、对象-续集**](http://blog.csdn.net/lvbey/article/details/53230978)

标签： [java](http://www.csdn.net/tag/java)[j2ee](http://www.csdn.net/tag/j2ee)[Spring](http://www.csdn.net/tag/Spring)[Annotation](http://www.csdn.net/tag/Annotation)[注解](http://www.csdn.net/tag/%e6%b3%a8%e8%a7%a3)

2016-11-19 15:50 36人阅读 [评论](http://blog.csdn.net/lvbey/article/details/53230978#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/lvbey/article/details/53230978#report)

 分类：



编程语言/IT/网络/java（12）



版权声明：本文为博主原创文章，未经博主允许不得转载。

1、支持通过[**spring**](http://lib.csdn.net/base/javaee)的xml配置文件来制定要获取注解类的包，如何获取配置文件的参数：

**[java]** [view plain](http://blog.csdn.net/lvbey/article/details/53230978) [copy](http://blog.csdn.net/lvbey/article/details/53230978)



1. /\*\*
2. \* 通过依赖注入获取配置文件中的属性值
3. \* @param basePackages
4. \*/
5. @Resource
6. **public** **void** setBasePackages(String... basePackages) {
7. **this**.basePackages = basePackages;
8. }

2、Spring的xml配置文件：

**[html]** [view plain](http://blog.csdn.net/lvbey/article/details/53230978) [copy](http://blog.csdn.net/lvbey/article/details/53230978)



1. **<bean** class="com.lvbey.service.ServiceBean" scope="singleton"**>**
2. <!-- 自动扫描的包 -->
3. **<property** name="basePackages"**>**
4. **<list>**
5. **<value>**com.xxxx.service**</value>**
6. **<value>**com.wwww.service**</value>**
7. **</list>**
8. **</property>**
9. **</bean>**

3、由于在某些情况下，我们不想把拥有@Service的某些类(少数部分)获取出来，我们可以加上一个自定义注解：@NotAutowired2Service

所以，我们需要自己写一个自定义的注解类：

**[java]** [view plain](http://blog.csdn.net/lvbey/article/details/53230978) [copy](http://blog.csdn.net/lvbey/article/details/53230978)



1. /\*\*
2. \* 在某个类上添加了该注解后，将不会自动注入到需要被获取的service中
3. \* @author Lvbey
4. \*/
5. @Target({ElementType.TYPE})
6. @Retention(RetentionPolicy.RUNTIME)
7. **public** **@interface** NotAutowired2Service {
8. }

好了，现在我们支持从配置文件获取指定的包，支持自定义注解了，说了这么多，直接上源代码：

**[java]** [view plain](http://blog.csdn.net/lvbey/article/details/53230978) [copy](http://blog.csdn.net/lvbey/article/details/53230978)



1. **package** com.lvbey.service;
3. **import** java.util.HashMap;
4. **import** java.util.Map;
6. **import** javax.annotation.Resource;
8. **import** org.osgi.service.component.annotations.Component;
9. **import** org.springframework.beans.BeansException;
10. **import** org.springframework.context.ApplicationContext;
11. **import** org.springframework.context.ApplicationContextAware;
12. **import** org.springframework.context.annotation.Scope;
14. **import** com.lvbey.service.util.ClassUtil;

17. /\*\*
18. \* @author Lvbey
19. \*/
20. @Component //交给spring管理方便其他类获取该类对象
21. @Scope(value="singleto")//单例
22. **public** **class** ServiceBean **implements** ApplicationContextAware{
24. **private** ApplicationContext applicationContext;
26. /\*\*
27. \* 获取到的类的实例对象
28. \*/
29. **private** Map<String, Object> classInstances;

32. /\*\*
33. \* 待扫描的基础包名
34. \*/
35. **private** String[] basePackages;

38. /\*\*
39. \* 通过依赖注入获取配置文件中的属性值
40. \* @param basePackages
41. \*/
42. @Resource
43. **public** **void** setBasePackages(String... basePackages) {
44. **this**.basePackages = basePackages;
45. }
47. @Override
48. **public** **void** setApplicationContext(ApplicationContext applicationContext) **throws** BeansException {
49. **this**.applicationContext = applicationContext;
50. }

53. **public** **void** init(){
54. **if**(**this**.applicationContext == **null**){
55. **return**;
56. }
57. **if**(**this**.classInstances == **null**){
58. **this**.classInstances = **new** HashMap<String,Object>();
59. }
60. //获取有NotAutowired2Service注解的实例
61. Map<String, Object> beansWithNotAutowired2ServiceMap = **this**.applicationContext.getBeansWithAnnotation(com.lvbey.service.annotation.NotAutowired2Service.**class**);
63. Map<String, Object> beansWithAnnotationMap = **this**.applicationContext.getBeansWithAnnotation(org.springframework.stereotype.Service.**class**);
65. Class<? **extends** Object> clazz = **null**;
66. **for**(Map.Entry<String, Object> entry : beansWithAnnotationMap.entrySet()){
67. clazz = entry.getValue().getClass();//获取到实例对象的class信息
68. Class<? **extends** Object>  [] interfaces = clazz.getInterfaces();
69. **for**(Class<? **extends** Object>  aInterface : interfaces){
70. String aInterfaceName = aInterface.getName();//接口的完整名
71. **for**(String packageName : **this**.basePackages){
72. **if**(aInterfaceName.startsWith(packageName)){//如果这个接口是在指定的包下
74. //接口实例名(只是将接口的首字母换成小写)
75. String aInterfaceSimpleName = ClassUtil.getDefaultInstanceName(aInterface);
77. //如果这个接口没有NotServiceBean注解
78. **if**(beansWithNotAutowired2ServiceMap.containsValue(entry.getValue())){
79. System.out.println(entry.getValue() + " has NotAutowired2Service Annotation");
80. }**else**{
81. classInstances.put(aInterfaceSimpleName, entry.getValue());
82. }
83. }
84. }
85. }
86. }
87. }
89. **public** Map<String, Object> getClassInstances() {
90. **if**(**this**.classInstances == **null**){
91. init();
92. }
93. **return** **this**.classInstances;
94. }
95. }

里面用到了一个util工具类：

**[java]** [view plain](http://blog.csdn.net/lvbey/article/details/53230978) [copy](http://blog.csdn.net/lvbey/article/details/53230978)



1. **public** **class** ClassUtil {
2. /\*\*
3. \* 根据这个类来获取默认的实例名（默认：将首字母换成小写）
4. \*
5. \* @param clazz
6. \*            类信息
7. \* @return 默认的实例名
8. \*/
9. **public** **static** String getDefaultInstanceName(Class<?> clazz) {
10. **if** (clazz == **null**) {
11. **return** **null**;
12. }
13. String className = clazz.getSimpleName();
14. String firsrLowerChar = className.substring(0, 1).toLowerCase();
15. className = firsrLowerChar + className.substring(1);
16. **return** className;
17. }
18. }

然后怎么使用这个类呢？很简单，在你需要使用的那个类里面加上Spring的自动注入就行：

**[java]** [view plain](http://blog.csdn.net/lvbey/article/details/53230978) [copy](http://blog.csdn.net/lvbey/article/details/53230978)



1. @Autowired
2. **private** ServiceBean serviceBean;
3. serviceBean.getClassInstances() ;//获取得到的map集合,剩下的想干啥干啥

对于不想写代码，想直接使用的看官。。。。。。轻戳[下载源码.jar](http://download.csdn.net/download/lvbey/9687411)

**[java]** [view plain](http://blog.csdn.net/lvbey/article/details/53230978) [copy](http://blog.csdn.net/lvbey/article/details/53230978)



1. <pre code\_snippet\_id="1994379" snippet\_file\_name="blog\_20161119\_4\_8531241"></pre>
2. <pre></pre>