[**java中借助Spring获取所有带有指定注解的接口、类、对象**](http://blog.csdn.net/lvbey/article/details/53230334)

标签： [J2EE](http://www.csdn.net/tag/J2EE)[Spring](http://www.csdn.net/tag/Spring)[注解](http://www.csdn.net/tag/%e6%b3%a8%e8%a7%a3)[java](http://www.csdn.net/tag/java)[Annotation](http://www.csdn.net/tag/Annotation)

2016-11-19 15:34 46人阅读 [评论](http://blog.csdn.net/lvbey/article/details/53230334#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/lvbey/article/details/53230334#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。

最近因为遇到有这个需求，花了两天的时间把问题解决了。

具体需求是这样的：由于公司框架需要，需要使用@Autowired注解将所有的@service类自动注入到一个指定的类。但是有个同事新添加了一个Service，忘了手动添加@Autowired注解注入，导致所有使用这个Service的功能都瘫痪了，由于是两个项目相互依赖，他没有检查出来这个问题。后来我找到了问题也解决了，但是！！！！你们是没看那个@Autowired的注入代码。我们这个项目有大约50个Service，在添加引用和注册对象就需要100行代码才能搞定，而且还可能像我同事一样忘记手动添加@Autowired注解来注入，所以后来我就根据这个需求写了一下这个功能。

需求名称可以简单描述为：获取所有带有指定注解的接口、类、对象。现在基本上都是使用[**spring**](http://lib.csdn.net/base/javaee)来管理对象了嘛，所以这篇文章是建立在Spring的一些基础上的。进入正题：

公用的代码片段如下：

1、我们需要获得ApplicationContext 的对象。

方法：实现ApplicationContextAware接口，实现setApplicationContext()这个方法即可，它会将ApplicationContext 的对象传递过来，我们在类中添加引用即可。

**[java]** [view plain](http://blog.csdn.net/lvbey/article/details/53230334) [copy](http://blog.csdn.net/lvbey/article/details/53230334)



1. @Override
2. **public** **void** setApplicationContext(ApplicationContext applicationContext) **throws** BeansException {
3. **this**.applicationContext = applicationContext;
4. }

2、通过applicationContext的getBeansWithAnnotation()获取所有指定注解的bean。

**[java]** [view plain](http://blog.csdn.net/lvbey/article/details/53230334) [copy](http://blog.csdn.net/lvbey/article/details/53230334)



1. Map<String, Object> beansWithAnnotationMap = **this**.applicationContext.getBeansWithAnnotation(org.springframework.stereotype.Service.**class**);

3、获取到带有注解的bean Map后，我们可以遍历这个map获取每个bean的接口、类：

**[java]** [view plain](http://blog.csdn.net/lvbey/article/details/53230334) [copy](http://blog.csdn.net/lvbey/article/details/53230334)



1. Class<? **extends** Object> clazz = **null**;
2. **for**(Map.Entry<String, Object> entry : beansWithAnnotationMap.entrySet()){
3. clazz = entry.getValue().getClass();//获取到实例对象的class信息
4. Class<? **extends** Object>  [] interfaces = clazz.getInterfaces();
5. **for**(Class<? **extends** Object>  aInterface : interfaces){
6. //接口信息
7. }
8. }

以上就是具有公用意义的代码段，我待会将完整的源代码发出来，在下一篇文章中，该功能支持Annotation注解以及spring的xml配置文件指定包。

[下一篇的传送门](http://blog.csdn.net/lvbey/article/details/53230978)。