<http://blog.csdn.net/sd4000784/article/details/7745947>

版权声明：本文为博主原创文章，未经博主允许不得转载。

以前框架使用quartz框架执行定时调度问题、

老大说这配置太麻烦、每个调度都需要多加在[**spring**](http://lib.csdn.net/base/javaee)的配置中、

能不能减少配置的量从而提高开发效率、

最近看了看spring的 scheduled的使用注解的方式进行调度、

感觉很方便、起码配置的东西少了很多、

所以留下来以备忘了、

首先要配置我们的spring.xml

xmlns 多加下面的内容、

**[html]** [view plain](http://blog.csdn.net/sd4000784/article/details/7745947) [copy](http://blog.csdn.net/sd4000784/article/details/7745947)

1. xmlns:task="http://www.springframework.org/schema/task"

然后xsi:schemaLocation多加下面的内容、

**[html]** [view plain](http://blog.csdn.net/sd4000784/article/details/7745947) [copy](http://blog.csdn.net/sd4000784/article/details/7745947)

1. http://www.springframework.org/schema/task
2. http://www.springframework.org/schema/task/spring-task-3.1.xsd

最后是我们的task任务扫描注解

**[html]** [view plain](http://blog.csdn.net/sd4000784/article/details/7745947) [copy](http://blog.csdn.net/sd4000784/article/details/7745947)

1. **<task:annotation-driven/>**

我的配置扫描位置是：

**[html]** [view plain](http://blog.csdn.net/sd4000784/article/details/7745947) [copy](http://blog.csdn.net/sd4000784/article/details/7745947)

1. **<context:annotation-config/>**
2. **<bean** class="org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor"**/>**
3. **<context:component-scan** base-package="com.test"**/>**

扫描的是com.test这样的包下的内容、

下面需要接口和实现（我的这几个**[Java](http://lib.csdn.net/base/java" \o "Java 知识库" \t "_blank)**文件都是com.test的包下的、）

**[java]** [view plain](http://blog.csdn.net/sd4000784/article/details/7745947) [copy](http://blog.csdn.net/sd4000784/article/details/7745947)

1. **public** **interface** IMyTestService {
2. **public** **void** myTest();
3. }

**[java]** [view plain](http://blog.csdn.net/sd4000784/article/details/7745947) [copy](http://blog.csdn.net/sd4000784/article/details/7745947)

1. @Component  //import org.springframework.stereotype.Component;
2. **public** **class** MyTestServiceImpl  **implements** IMyTestService {
3. @Scheduled(cron="0/5 \* \*  \* \* ? ")   //每5秒执行一次
4. @Override
5. **public** **void** myTest(){
6. System.out.println("进入测试");
7. }
8. }

执行后控制台就会打印出   进入测试   了

需要注意的几点：

1、spring的@Scheduled注解  需要写在实现上、

2、 定时器的任务方法不能有返回值（如果有返回值，spring初始化的时候会告诉你有个错误、需要设定一个proxytargetclass的某个值为true、具体就去百度google吧）

3、实现类上要有组件的注解@Component

剩下的就是corn表达式了、具体使用以及参数请百度google、

下面只例出几个式子

CRON表达式    含义   
"0 0 12 \* \* ?"    每天中午十二点触发   
"0 15 10 ? \* \*"    每天早上10：15触发   
"0 15 10 \* \* ?"    每天早上10：15触发   
"0 15 10 \* \* ? \*"    每天早上10：15触发   
"0 15 10 \* \* ? 2005"    2005年的每天早上10：15触发   
"0 \* 14 \* \* ?"    每天从下午2点开始到2点59分每分钟一次触发   
"0 0/5 14 \* \* ?"    每天从下午2点开始到2：55分结束每5分钟一次触发   
"0 0/5 14,18 \* \* ?"    每天的下午2点至2：55和6点至6点55分两个时间段内每5分钟一次触发   
"0 0-5 14 \* \* ?"    每天14:00至14:05每分钟一次触发   
"0 10,44 14 ? 3 WED"    三月的每周三的14：10和14：44触发   
"0 15 10 ? \* MON-FRI"    每个周一、周二、周三、周四、周五的10：15触发