처리결과 : 성공

**Spring Task Scheduler 손쉽게 설정하기**

**ref :** <http://webprogramer.kr/blog/P000000289/post.do>

**Spring Task Scheduler 손쉽게 설정하기**

예전에 회사를 다니면서 했던 작업중에 주기적으로 DB를 백업하거나 특정시간이

되면 타 시스템에 접속하여 데이터를 가져오거나 새벽 시간에 메일을 보낸다거나 할때

사용했던것이**Linux**에 **Cron**(크론탭 이라고했음) 이였다. 크론탭을 등록하고 Java로

프로그램 하여 처리 했었는데 크론탭을 설정할때 java path설정하고 java lib를 로드하고

하려면 조금은 번거로웠었다. Spring에 Timer나 Quartz가 있지만, 내가 봤을때 손쉽게

설정하고 써먹기 좋은게 Task Scheduler 라서 사용방법을 알아보도록 하겠다.

**1. XML설정**

 - 현재 egov-com-servlet.xml 설정 일부

   (옵션이 있지만 옵션은 검색을 통해서 찾아보기 바란다.)

1. <beans xmlns="http://www.springframework.org/schema/beans"
2. xmlns:context="http://www.springframework.org/schema/context"
3. xmlns:p="http://www.springframework.org/schema/p"
4. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5. xmlns:mvc="http://www.springframework.org/schema/mvc"
6. xmlns:task="http://www.springframework.org/schema/task"
7. xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
8. http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.0.xsd
9. http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd
10. http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-4.0.xsd">
11. <context:component-scan base-package="egovframework">
12. <context:include-filter expression="org.springframework.stereotype.Controller" type="annotation"/>
13. <context:exclude-filter expression="org.springframework.stereotype.Service" type="annotation"/>
14. <context:exclude-filter expression="org.springframework.stereotype.Repository" type="annotation"/>
15. </context:component-scan>
17. <!-- Activates @Scheduled and @Async annotations for scheduling -->
18. <task:annotation-driven />

**2. java 코드 작성**

- 3초마다 실행되게 설정

 (크론 expression 설정방법은 하단에 기재)

1. @Component
2. public class BlogTask {
4. @Resource(name="tagService")
5. private TagService tagService;
7. @Resource(name="counterService")
8. private CounterService counterService;
10. private static final Logger LOGGER = LoggerFactory.getLogger(BlogTask.class);
12. /\*\*
13. \* @desc : Task Scheduler Test
14. \*/
15. @Scheduled(cron="0/3 \* \* \* \* \*")
16. public void taskSchedulerTest() {
17. LOGGER.debug("taskSchedulerTest - TASK START ");
18. try {
20. LOGGER.debug("taskSchedulerTest - Job!!");
22. LOGGER.debug("taskSchedulerTest - Succ");
23. } catch (Exception ex) {
24. ex.printStackTrace();
25. LOGGER.debug("taskSchedulerTest - Fail");
26. }
27. LOGGER.debug("taskSchedulerTest - TASK END ");
28. }
29. }

- 로그확인 3초마다 실행된 로그가 찍혔다.

1. 2016/08/23 10:33:48,001 [pool-1-thread-1] DEBUG egovframework.blog.task.BlogTask - taskSchedulerTest - TASK START
2. 2016/08/23 10:33:48,001 [pool-1-thread-1] DEBUG egovframework.blog.task.BlogTask - taskSchedulerTest - Job!!
3. 2016/08/23 10:33:48,001 [pool-1-thread-1] DEBUG egovframework.blog.task.BlogTask - taskSchedulerTest - Succ
4. 2016/08/23 10:33:48,001 [pool-1-thread-1] DEBUG egovframework.blog.task.BlogTask - taskSchedulerTest - TASK END
5. 2016/08/23 10:33:51,001 [pool-1-thread-1] DEBUG egovframework.blog.task.BlogTask - taskSchedulerTest - TASK START
6. 2016/08/23 10:33:51,001 [pool-1-thread-1] DEBUG egovframework.blog.task.BlogTask - taskSchedulerTest - Job!!
7. 2016/08/23 10:33:51,001 [pool-1-thread-1] DEBUG egovframework.blog.task.BlogTask - taskSchedulerTest - Succ
8. 2016/08/23 10:33:51,001 [pool-1-thread-1] DEBUG egovframework.blog.task.BlogTask - taskSchedulerTest - TASK END
9. 2016/08/23 10:33:54,001 [pool-1-thread-1] DEBUG egovframework.blog.task.BlogTask - taskSchedulerTest - TASK START
10. 2016/08/23 10:33:54,001 [pool-1-thread-1] DEBUG egovframework.blog.task.BlogTask - taskSchedulerTest - Job!!
11. 2016/08/23 10:33:54,001 [pool-1-thread-1] DEBUG egovframework.blog.task.BlogTask - taskSchedulerTest - Succ
12. 2016/08/23 10:33:54,001 [pool-1-thread-1] DEBUG egovframework.blog.task.BlogTask - taskSchedulerTest - TASK END

이제 쉽게 스케쥴러를 사용하실 수 있습니다.

[참고] Cron 표현식

|  |  |  |
| --- | --- | --- |
| 순서 | 필드명 | 사용 가능한 값 |
| 1 | seconds | 0~59 , - \* / |
| 2 | minutes | 0~59 , - \* / |
| 3 | hours | 0~23 , - \* / |
| 4 | day of month | 1~31 , - \* ? / L W |
| 5 | month | 1~12 or JAN-DEC , - \* / |
| 6 | day of week | 1-7 or SUN-SAT , - \* ? / L # |
| 7 | years (optional) | 1970~2099 , - \* / |

특수문자의 의미

|  |  |  |
| --- | --- | --- |
| 기호 | 의미 | 사용 예 |
| \* | 모든 수를 의미 | seconds에서 사용하면 매초, minutes에서 사용하면 매분, hours에서 사용하면 매시간 |
| ? | 해당 항목을 사용하지 않음 | day of month에서 사용하면 월중 날짜를 지정하지 않음. day of week에서 사용하면 주중 요일을 지정하지 않음 |
| - | 기간을 설정 | hours에서 10-12이면 10시, 11시, 12시에 동작  minutes에서 58-59이면 58분, 59분에 동작 |
| , | 특정 시간을 지정 | day of week에서 2,4,6이면 월,수,금에만 동작함 |
| / | 시작시간과 반복 간격 설정 | seconds위치에 0/15로 설정하면 0초에 시작해서 15초 간격으로 동작  minutes위치에 5/10으로 설정하면 5분에 시작해서 10분 간격으로 동작 |
| L | 마지막 기간에 동작  day of month, day of week에서만 사용 | day of month에서 사용하면 해당월 마지막 날에 수행  day of week에서 사용하면 토요일에 수행 |
| W | 가장 가까운 평일 동작  day of month에만 사용 | 15W로 설정하면 15일이 토요일이면 가장 가까운 14일 금요일에 실행  15W로 설정하고 15일이 일요일이면 16일에 실행  15W로 설정하고 15일이 평일이면 15일에 실행 |
| LW | L과 W의 조합 | 그달의 마지막 평일에 동작 |
| # | 몇 번째 주와 요일 설정  day of week에 사용 | 6#3이면 3 번째 주 금요일에 동작  4#2이면 2번째 주 수요일에 동작 |

사용 예

|  |  |
| --- | --- |
| 표현식 | 의미 |
| 0 0 12 \* \* \* | 매일 12시에 실행 |
| 0 15 10 \* \* \* | 매일 10시 15분에 실행 |
| 0 \* 14 \* \* \* | 매일 14시에 0분~59분까지 매분 실행 |
| 0 0/5 14 \* \* \* | 매일 14시에 시작해서 5분 간격으로 실행 |
| 0 0/5 14,18 \* \* \* | 매일 14시, 18시에 시작해서 5분 간격으로 실행 |
| 0 0-5 14 \* \* \* | 매일 14시에 0분, 1분, 2분, 3분, 4분, 5분에 실행 |
| 0 0 20 ? \* MON-FRI | 월~금일 20시 0분 0초에 실행 |
| 0 0/5 14 \* \* ? | 아무요일, 매월, 매일 14:00부터 14:05분까지 매분 0초 실행 (6번 실행됨) |
| 0 15 10 ? \* 6L | 매월 마지막 금요일 아무날이나 10:15:00에 실행 |
| 0 15 10 15 \* ? | 아무요일, 매월 15일 10:15:00에 실행 |
| \* /1 \* \* \* \* | 매 1분마다 실행 |
| \* /10 \* \* \* \* | 매 10분마다 실행 |

이전 Task와 Delay 설정

|  |  |
| --- | --- |
| 표현 | 의미 |
| fixed-delay | 이전에 실행된 task의 종료시간으로부터 정의된 시간만큼 지난 후 다음 task를 실행  밀리세컨드 단위 |
| fixed-rate | 이전에 실행된 task의 시작 시간으로부터 정의된 시간만큼 지난 후 다음 task를 실행  밀리세컨드 단위  [출처] [Sprin](http://blog.naver.com/lovemema/140200056062" \t "_blank) |

**태그**