크론정리

- description : 크론정리
- author : 오션
- email : shlim@repia.com
- lastupdate : 2022-06-30 Thu

크론정리

getNextValidTimeAfter()

Quartz 사용 시 cron에 정의한 시간에 정상 동작하는지를 확인해야 하는 경우, Quartz에서 제공하는 CronExpression클래스를 이용.

```
@Test
public void cronTimeTest() throws ParseException {
    // 매일 9, 12, 18시 정각에 스케줄링 동작
    String cronExample = "0 0 9,12,18 * * ?";

    CronExpression cronExpression = new CronExpression(cronExample);
    log.debug("cronExpression=[{}]", cronExpression);
    //cronExpression=[0 0 9,12,18 * * ?]

    Date nextTime1 = cronExpression.getNextValidTimeAfter(new Date());
    log.debug("nextTime1=[{}]", nextTime1);
    //nextTime1=[2022-07-01T09:00:00.000+0900]

    Date nextTime2 = cronExpression.getNextValidTimeAfter(nextTime1);
    log.debug("nextTime2=[{}]", nextTime2);
    //nextTime2=[2022-07-01T12:00:00.000+0900]

    Date nextTime3 = cronExpression.getNextValidTimeAfter(nextTime2);
    log.debug("nextTime3=[{}]", nextTime3);
    //nextTime3=[2022-07-01T18:00:00.000+0900]

    Date nextTime4 = cronExpression.getNextValidTimeAfter(nextTime3);
    log.debug("nextTime4=[{}]", nextTime4);
    //nextTime4=[2022-07-02T09:00:00.000+0900]

    boolean compareResult = nextTime2.equals(nextTime3);
    assertEquals(compareResult, true);
}
```

```
@Test
public void 월요정기회의() throws Exception {
    String cronEx = "0 59 23 ? * 2";
    int dayOffset = 2;

    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyyMMdd",
Locale.getDefault());

    Calendar cal = Calendar.getInstance();
    cal.setTime(new Date());
    log.debug("cal.getTime()=[{}]", cal.getTime());
    //현재 일시를 cal에 셋팅
    //cal.getTime()=[2022-07-01T10:52:04.257+0900]

    cal.add(Calendar.DATE, dayOffset);
    log.debug("dayOffset=[{}], cal.getTime() after add dayOffset=[{}]",
dayOffset, cal.getTime());
    // 현재 일시에 dayOffset을 더한 결과
    //dayOffset=[2], cal.getTime() after add
dayOffset=[2022-07-03T10:52:04.257+0900]

    CronExpression cronExpression = new CronExpression(cronEx);
    log.debug("cronExpression=[{}]", cronExpression);
    //String으로 입력된 cronEx를 크론표현식으로 변환
    //cronExpression=[0 59 23 ? * 2]

    Date cronDate = cronExpression.getNextValidTimeAfter(cal.getTime());
    log.debug("cronDate=[{}]", cronDate);
    //(현재 일시+dayOffset)의 다음에 동작하는 크론 일시 표현
    //cronDate=[2022-07-04T23:59:00.000+0900]

    String cronDateStr = dateFormat.format(cronDate);
    log.debug("cronDateStr=[{}]", cronDateStr);
    //(현재 일시+dayOffset)의 다음에 동작하는 크론 일시를 위에서 정의한 yyyyMMdd형식으로
출력
    //cronDateStr=[20220704]

    String compareDateStr = dateFormat.format(cal.getTime());
    log.debug("compareDateStr=[{}]", compareDateStr);
    //(현재 일시+dayoffset)를 위에서 정의한 yyyyMMdd형식으로 출력
    //compareDateStr=[20220703]

    log.debug("cronDateStr=[{}] <<>> compareDateStr=[{}]", cronDateStr,
compareDateStr);
    //cronDateStr=[20220704] <<>> compareDateStr=[20220703]

    boolean compareResult = cronDateStr.equals(compareDateStr);
    //boolean ret = CrontabUtil.isMatchCronExpression(cronExpression,
dayOffset);
```

```
    assertEquals(compareResult, true);    // Failure
}
```

Ref LInk

CronExpression Test하기

오션,, cron,, 크론