Spring Batch – Timing Configurations / Improved Cron Expressions

Usage

You typically create cron triggers with the @Scheduled annotation, which uses CronExpression internally, as of Spring Framework 5.3. This means that you can already start using the New Features if you are on that version.

If you want to play around with CronExpression yourself, you can create one through the static parse method:
[crayon-6684a683ccd43189283344/]
[crayon-6684a683ccd4b931853623/]

Every 5 Seconds Example
[crayon-6684a683ccd4d685001870/]

Some rules apply:

- A field may be an asterisk (*), which always stands for “first-last”. For the day-of-the-month or day-of-the-week fields, a question mark (?) may be used instead of an asterisk.
- Commas (,) are used to separate items of a list.
- Two numbers separated with a hyphen (-) express a range of numbers. The specified range is inclusive.
- Following a range (or *) with / specifies the interval of the number’s value through the range.
- English names can also be used for the day-of-month and day-of-week fields. Use the first three letters of the particular day or month (case does not matter).

Here are some examples:

Cron Expression	Meaning
0 0 * * * *	top of every hour of every day
*/10 * * * * *	every ten seconds
0 0 8-10 * * *	8, 9 and 10 o’clock of every day
0 0 6,19 * * *	6:00 AM and 7:00 PM every day
0 0/30 8-10 * * *	8:00, 8:30, 9:00, 9:30, 10:00 and 10:30 every day
0 0 9-17 * * MON-FRI	on the hour nine-to-five weekdays
0 0 0 25 12 ?	every Christmas Day at midnight

The next method returns the next occurrence of the trigger or null if there is none. It takes a java.time.temporal.Temporal as a parameter, which means it accepts not only LocalDateTime but also ZonedDateTime if time-zones are relevant.

New Features

Using the java.time APIs let us introduce several new features that put Spring’s support for cron expressions on an equal footing with other schedulers. You can start using these features in @Scheduled as of Spring Framework 5.3.

Macros

Expressions such as 0 0 * * * * are hard for humans to parse and are, therefore, hard to fix in case of bugs. To improve readability, Spring now supports the following macros, which represent commonly used sequences. You can use these macros instead of the six-digit value, thus: @Scheduled(cron = "@hourly").

Macro	Meaning
@yearly (or @annually)	once a year (0 0 0 1 1 *)

Macro	Meaning
@monthly	once a month (0 0 0 1 * *)
@weekly	once a week (0 0 0 * * 0)
@daily (or @midnight)	once a day (0 0 0 * * *), or
@hourly	once an hour, (0 0 * * * *)

_

Ref : https://spring.io/blog/2020/11/10/new-in-spring-5-3-improved-cron-expressions

Ref : https://www.baeldung.com/spring-batch-start-stop-job

_