<minute> <hour> <day-of-month> <month> <day-of-week> <command>

(Hiện tại đang conflict khi chạy thực tế, tham số đầu tiên là s(giây))  
ex: **At 12:00 p.m. (noon) every day**:

0 12 \* \* ? Copy

**Every five minutes starting at 1 p.m. and ending at 1:55 p.m. and then starting at 6 p.m. and ending at 6:55 p.m., every day**:

0/5 13,18 \* \* ?Copy

**Every minute starting at 1 p.m. and ending at 1:05 p.m., every day**:

0-5 13 \* \* ?Copy

**At 1:15 p.m. and 1:45 p.m. every Tuesday in the month of June**:

15,45 13 ? 6 TueCopy

**At 9:30 a.m. every Monday, Tuesday, Wednesday, Thursday and Friday**:

30 9 ? \* MON-FRICopy

**At 9:30 a.m. on the 15th day of every month**:

30 9 15 \* ?Copy

**At 6 p.m. on the last day of every month**:

0 18 L \* ?Copy

**At 6 p.m. on the third to last day of every month**:

0 18 L-3 \* ?Copy

**At 10:30 a.m. on the last Thursday of every month**:

30 10 ? \* 5LCopy

**At 10 a.m. on the third Monday of every month**:

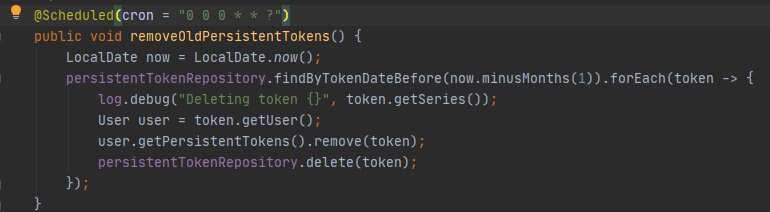
0 10 ? \* 2#3Copy

**At 12 midnight on every day for five days starting on the 10th day of the month**:

0 0 10/5 \* ?

- cron expression: [A Guide To Cron Expressions | Baeldung](https://www.baeldung.com/cron-expressions)

**\* Scheduler:**Sử dụng scheduler để lên lịch trình cho các task trong ứng dụng có thẻ có nhiều task chạy trong 1 ứng dụngs, nhưng nếu có nhiều instance của task, sẽ không đảm bảo được sự đồng nhất khi xử lý các task, solution là sd: **shedlock**



**\* Shedlock**  
- sd shedLock để run job in distributed system: <https://medium0.com/@pinkijindal786/distributed-lock-for-scheduled-tasks-shedlock-using-springboot-27838c7aace9>

Vd:

@Scheduled(cron = "0/5 \* \* \* \* ?")  
@SchedulerLock(name = "app1", lockAtLeastFor = "2s", lockAtMostFor = "120s")  
public void runSchedulerLock(){  
 logger.info("Running sheduler-lock app1-1-1-1-1: " + appName);  
}  
  
@Scheduled(cron = "0/5 \* \* \* \* ?")  
@SchedulerLock(name = "app1", lockAtLeastFor = "2s", lockAtMostFor = "120s")  
public void runSchedulerLock1(){  
 logger.info("Running sheduler-lock app2-2-2-2-2: " + appName);  
}  
  
/\*  
phase1  
\* two above schedule-lock have the same name, and shedlock while controll them through table: shedlock  
\* in table: shedlock, if lock\_until > now of name app1, then there is 1 task running (1 in 2 above task) and it holds the lock,  
\* so the left over can not run its execution until the lock get released  
\* phase2:  
when the lock get released, which task get the lock will repeat this cycle again  
  
the same thing happen when we have multiple instance of application and they have the same task, we should use shed lock like  
this to prevent simutanious running.  
  
- the tasks name x from many instances of services are ready to run when in shedlock table, the of of name x has value of  
lock\_until <= now()  
  
- param: lockAtLeastFor is useful for short period running task, because, when task execution finish in no time, other insance can run right away  
with the unconsistent data. bacause the clock among instaces are a little different from each other  
 . if the running task finished earlier than value: lock\_at + lockAtLeastFor, then the lock\_until will be set to (lock\_at + lockAtLeastFor)  
 instead of now(), if running task finish after (lock\_at + lockAtLeastFor) then lock\_until will be set to now();  
- param: lockAtMostFor X: is very useful when running instace die or get suspended, the the lock will be released after X anyway  
then other instace can continue to execute task.  
\* \*/

Vd:

**Instance 1:**

2022-12-28 17:11:30.030 INFO 18996 --- [ scheduling-1] o.s.scheduling.config.ScheduledTask : Running sheduler-lock app1-1-1-1-1: test-swagger

**2022-12-28 17:11:35**.031 INFO 18996 --- [ scheduling-1] o.s.scheduling.config.ScheduledTask : Running sheduler-lock app1-1-1-1-1: test-swagger

2022-12-28 17:12:10.031 INFO 18996 --- [ scheduling-1] o.s.scheduling.config.ScheduledTask : Running sheduler-lock app1-1-1-1-1: test-swagger

**Instance 2:**

2022-12-28 17:11:40.030 INFO 12572 --- [ scheduling-1] o.s.scheduling.config.ScheduledTask : Running sheduler-lock app1-1-1-1-1: test-swagger

2022-12-28 17:11:45.032 INFO 12572 --- [ scheduling-1] o.s.scheduling.config.ScheduledTask : Running sheduler-lock app1-1-1-1-1: test-swagger

2022-12-28 17:11:50.035 INFO 12572 --- [ scheduling-1] o.s.scheduling.config.ScheduledTask : Running sheduler-lock app1-1-1-1-1: test-swagger

2022-12-28 17:11:55.031 INFO 12572 --- [ scheduling-1] o.s.scheduling.config.ScheduledTask : Running sheduler-lock app1-1-1-1-1: test-swagger

2022-12-28 17:12:00.065 INFO 12572 --- [ scheduling-1] o.s.scheduling.config.ScheduledTask : Running sheduler-lock app1-1-1-1-1: test-swagger

2022-12-28 17:12:05.034 INFO 12572 --- [ scheduling-1] o.s.scheduling.config.ScheduledTask : Running sheduler-lock app1-1-1-1-1: test-swagger