Date : 01/01/2021
Spring Boot 9AM
Mr. RAGHU

Spring Boot : Scheduling

=> Task(method) execute in a loop (multiple)
=> Period Of Time , Point Of Time
=> To activate Scheduling , add : @EnableScheduling
=> At (new class and) method level , add : @Scheduled

*) @Scheduled contains 3 concepts.
a) fixedDelay = An Exact Time gap between method executions.
Last method finish time to next method start time.

b) fixedRate = Max time gap between method executions.

Case#1 if fixedRate > method Execution Time
Time Gap = fixedRate - method Execution Time

Case#2 if fixedRate <= method Execution Time
Time Gap = Zero.

--code--

```
package in.nareshit.raghu;

import java.util.Date;

import org.springframework.scheduling.annotation.Scheduled;
import org.springframework.stereotype.Component;

@Component
public class MessageService {

        @Scheduled(initialDelay = 4000,fixedRate = 1000)
        public void showMsg() {
                System.out.println("FROM MESSAGE " + new Date());
        }

}
```

---

c)** cron expression : default Point of Time, it support even period of time

SEC	MIN	HRS	DAY	MONTH	WEEK-DAY
0-59	0-59	0-23	1-31	1-12	SUN-SAT

=> We can even add few symbols in cron expressions
Those are:

* = all/any/every
, = possible values (multiple values)
- = Range (from - to)
/ = Period of Time
? = all/every/any [Applied to position only DAY and WEEK-DAY

when month is provided]

Ex#1
cron = "0 0 10 * * *"

=> Execute given task (method), Every day at 10:00:00 AM

Ex#2
cron= "0 0 22 * * *"

=> Execute given task (method), Every Day at 10:00:00 PM

Ex#3
cron= "0 30 11 * * *"

=> Execute given task (method), Every Day at 11:30:00 AM

Ex#4
cron= "0 0 9,10,11 * * *"

=> Execute given task (method), Every Day 3 times
9:00:00 AM
10:00:00 AM
11:00:00 AM

Ex#5
cron= "0 0 7-10 * * *"

=> Execute given task (method), Every Day 4 times
7:00:00 AM
8:00:00 AM
9:00:00 AM
10:00:00 AM

Ex#6
cron= "0 10 * * * *"

=> For every 10 mins (Wrong Answer)
9:45:00 App Started
9:55:00 (Next execution)
10:05:00 (Next execution)

=> Execute given task (method), For Every hour 10th Minute. (Right Answer)
9:45:00 App Started
10:10:00 (1st execution 10AM 10th Minute)
11:10:00 (Next execution 11AM 10th Minute)
12:10:00 (Next execution 12PM 10th Minute)
13:10:00 (Next execution 1PM   10th Minute)

Ex#7
cron="0 30 * * * *"

=> Execute given task (method), For Every hour 30th Minute. (Right Answer)

```
   9:45:00 App Started
   1st execution  -- 10:30:00 AM
   Next execution -- 11:30:00 AM
   Next execution -- 12:30:00 PM
   Next execution -- 1:30:00  PM


Ex#8
cron="10 * * * * *"

=> Every 10 sec gap (Wrong Answer).

=> Execute given task (method), For Every Minute 10th Sec. (Right
Answer)

   9:45:00 App Started
   1st execution  -- 9:45:10  (10th Sec of Minute)
   Next execution -- 9:46:10
   Next execution -- 9:47:10
   Next execution -- 9:48:10
   ..etc

       SEC    MIN    HRS    DAY    MONTH    WEEK-DAY
      0-59   0-59   0-23   1-31    1-12     SUN-SAT


---example--
package in.nareshit.raghu;

import java.util.Date;

import org.springframework.scheduling.annotation.Scheduled;
import org.springframework.stereotype.Component;

@Component
public class MessageService {

        @Scheduled(cron = "10 * * * * *")
        public void showMsg() {
                System.out.println("FROM MESSAGE " + new Date());
        }

}



Ex#9
cron="20 * * * *"

IllegalStateException: Encountered invalid @Scheduled method
'showMsg':
Cron expression must consist of 6 fields (found 5 in "20 * * * *")
------------------------------------------------------------------------
-------
```