Date : 31/12/2020
Spring Boot 9AM
Mr. RAGHU

Spring Boot : Scheduling

*) Scheduling :-
Executing a Task in a loop (multiple times), based on
Period Of Time (or) Point Of Time.

Period : Time gap
Ex: 5 mins, 6 hours, 3 days, 2 months,..etc

Point : Exact Date and Time
Ex: Mar 31st 9AM,
12th JAN 6PM,
..etc

Examples for Scheduling:-
a) Bank Account Statement (sent email)
b) Credit Card Billing
c) Reports for Business
d) EMI Payment details
e) Employee Salary Payment, PaySlip Generation
f) Daily class (ZoomMeeting)
g) Mobile Auto Recharge
h) Insurance Policy Payment
..etc

Base Factor : NANO SEC
1 MICRO = 1000 x NANO SEC;
1 MILLI = 1000 x MICRO SEC;
1 SECOND = 1000 x MILLI SEC;
1 MINUTE = 60 x SECOND;
1 HOUR   = 60 x MINUTE;
1 DAY   = 24 x HOUR;

Working with Scheduling

S#1 Our Spring Boot Application, will not activate Scheduling default.
(Every application may not need this).
To work with Scheduling add annotation : @EnableScheduling at starter class.

S#2 Define one class and method, Add annotation: @Scheduled over the method

*) @Scheduled is defined in 3 ways.
a) fixedDelay
b) fixedRate
c) *** cron (cron expression | cron jobs)

a) fixedDelay :
  This concept comes under period of time,
  it takes number (input) in Mill sec
  When application is started, then Spring container
  creates object and call method in Loop.

**) Normally web application are executed using Server.
  When Server is started , then Scheduling is started as background thread
  (daemon thread) which will not effect actual flow and use of application.
  When application is stopped then scheduling is stopped.

*) initialDelay (in mill sec) this is used to provide time gap between
   application startup and first method call.

---Code------------------

S#1 Create Spring Starter Project
Name : SpringBoot2SchedulingEx

S#2 Add Annotation over starter class: @EnableScheduling
S#3 Define one Schduler class

```
package in.nareshit.raghu;

import java.util.Date;

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

@Component
public class MessageService {

        //@Scheduled(fixedDelay = 120000) //2 mins
        @Scheduled(initialDelay = 3000,fixedDelay = 5000) // 5 sec
        public void showMsg() {
                System.out.println("FROM MESSAGE " + new Date());
        }

}
```

*) Note:
 If we define @Scheduled witout any value (or attribute)
 then -> IllegalStateException: Encountered invalid @Scheduled method 'showMsg':
 Exactly one of the 'cron', 'fixedDelay(String)', or
 'fixedRate(String)' attributes is required.

*) Providing -ve value to Scheduling is a invalid case
    @Scheduled(fixedDelay = -5555) (same exception as like above).
