#############################
What is Spring Scheduling ?
#############################

=> Spring scheduling is a feature provided by the Spring Framework that allows you to schedule tasks or methods to be executed at specific times or intervals.

=> It provides a convenient way to automate repetitive tasks, background processing, or any other operations that need to be executed periodically.

Spring scheduling is typically used in applications where you need to perform tasks at regular intervals, such as:

1) Generating reports or performing data backups at specific times.

2) Sending periodic notifications or reminders.

3) Updating cache or performing data synchronization.

4) Running background jobs or batch processing.

The scheduling functionality in Spring is achieved through the use of the @Scheduled annotation. By annotating a method with @Scheduled and configuring the scheduling properties, you can specify when and how often the method should be executed.

====================
what is fixedRate ?
====================

fixedRate: This attribute specifies the interval between the start times of consecutive task executions. It means that the task will be executed at a fixed rate, regardless of the duration of the previous task execution. For example, if you set fixedRate to 5000 milliseconds, the task will be executed every 5 seconds, regardless of how long the previous execution took. If the task execution takes longer than the specified interval, subsequent executions will be delayed, and the tasks may overlap if they take longer than the interval.

====================
what is fixedDelay ?
====================

fixedDelay : This attribute specifies the delay between the completion of the previous task execution and the start of the next execution. It means that the task will be executed with a fixed delay after the completion of the previous execution. For example, if you set fixedDelay to 5000 milliseconds, the task will be executed every 5 seconds after the previous execution completes. This ensures that there is a fixed delay between the completion of one execution and the start of the next, regardless of how long the task execution takes. If the task takes longer than the specified delay, the next execution will be scheduled immediately after the completion of the previous one.

==========================
What is cron expression ?
==========================

In Spring, cron expressions are commonly used to define the scheduling patterns for tasks or methods using the @Scheduled annotation. Cron expressions provide a flexible and powerful way to specify the timing of task executions. Here's an explanation of the cron expression format used in Spring:

A cron expression consists of six fields, representing different aspects of the schedule. The fields are separated by spaces:

â”Œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€ minute (0 - 59)
â”‚ â”Œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€ hour (0 - 23)
â”‚ â”‚ â”Œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€ day of the month (1 - 31)
â”‚ â”‚ â”‚ â”Œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€ month (1 - 12)
â”‚ â”‚ â”‚ â”‚ â”Œâ”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€ day of the week (0 - 7) (Sunday = 0 or
7)
â”‚ â”‚ â”‚ â”‚ â”‚
â”‚ â”‚ â”‚ â”‚ â”‚
* * * * * *

=====================================
Here's a breakdown of the fields:
====================================
Minute (0 - 59): Specifies the minute at which the task should execute.

Hour (0 - 23): Specifies the hour of the day at which the task should execute.

Day of the month (1 - 31): Specifies the day of the month at which the task should execute.

Month (1 - 12): Specifies the month of the year at which the task should execute.

Day of the week (0 - 7): Specifies the day of the week at which the task should execute. Both 0 and 7 represent Sunday.

Year (optional): Specifies the year in which the task should execute.

############################ https://crontab.cronhub.io/
######################################

Here are some examples of cron expressions:

0 * * * * *: Runs the task at the start of every minute.

0 0 * * * *: Runs the task at the start of every hour.

0 0 12 * * *: Runs the task at 12:00 PM (noon) every day.

0 0 8-18 * * *: Runs the task every hour between 8 AM and 6 PM.

0 0 0 1 * *: Runs the task at midnight on the first day of every month.

0 0 0 * * 1: Runs the task at midnight on every Sunday.

You can combine different values or use wildcards (*) to specify multiple values for a field. Additionally, you can use special characters such as - for ranges and / for increments.

Cron expressions provide a lot of flexibility to define complex schedules for your tasks. By customizing the fields appropriately, you can achieve precise scheduling according to your requirements.