@Slf4j

@Component

public class ScheduledService {

@Scheduled(cron = "0/5 \* \* \* \* \*")

public void scheduled(){

log.info("=====>>>>>使用cron {}",System.currentTimeMillis());

}

@Scheduled(fixedRate = 5000)

public void scheduled1() {

log.info("=====>>>>>使用fixedRate{}", System.currentTimeMillis());

}

@Scheduled(fixedDelay = 5000)

public void scheduled2() {

log.info("=====>>>>>fixedDelay{}",System.currentTimeMillis());

}

}

**执行时间的配置**

在上面的定时任务中，我们在方法上使用@Scheduled注解来设置任务的执行时间，并且使用三种属性配置方式：

1. fixedRate：定义一个按一定频率执行的定时任务
2. fixedDelay：定义一个按一定频率执行的定时任务，与上面不同的是，改属性可以配合initialDelay， 定义该任务延迟执行时间。
3. cron：通过表达式来配置任务执行时间

**cron表达式详解**

一个cron表达式有至少6个（也可能7个）有空格分隔的时间元素。按顺序依次为：

* 秒（0~59）
* 分钟（0~59）
* 3 小时（0~23）
* 4 天（0~31）
* 5 月（0~11）
* 6 星期（1~7 1=SUN 或 SUN，MON，TUE，WED，THU，FRI，SAT）
* 年份（1970－2099）

其中每个元素可以是一个值(如6),一个连续区间(9-12),一个间隔时间(8-18/4)(/表示每隔4小时),一个列表(1,3,5),通配符。由于”月份中的日期”和”星期中的日期”这两个元素互斥的,必须要对其中一个设置。配置实例：

* 每隔5秒执行一次：*/5*\* \* \* ?
* 每隔1分钟执行一次：0 */1*\* \* ?
* 0 0 10,14,16 \* \* ? 每天上午10点，下午2点，4点
* 0 0/30 9-17 \* \* ? 朝九晚五工作时间内每半小时
* 0 0 12 ? \* WED 表示每个星期三中午12点
* “0 0 12 \* \* ?” 每天中午12点触发
* “0 15 10 ? \* \*” 每天上午10:15触发
* “0 15 10 \* \* ?” 每天上午10:15触发
* “0 15 10 \* \* ? \*” 每天上午10:15触发
* “0 15 10 \* \* ? 2005” 2005年的每天上午10:15触发
* “0 \* 14 \* \* ?” 在每天下午2点到下午2:59期间的每1分钟触发
* “0 0/5 14 \* \* ?” 在每天下午2点到下午2:55期间的每5分钟触发
* “0 0/5 14,18 \* \* ?” 在每天下午2点到2:55期间和下午6点到6:55期间的每5分钟触发
* “0 0-5 14 \* \* ?” 在每天下午2点到下午2:05期间的每1分钟触发
* “0 10,44 14 ? 3 WED” 每年三月的星期三的下午2:10和2:44触发
* “0 15 10 ? \* MON-FRI” 周一至周五的上午10:15触发
* “0 15 10 15 \* ?” 每月15日上午10:15触发
* “0 15 10 L \* ?” 每月最后一日的上午10:15触发
* “0 15 10 ? \* 6L” 每月的最后一个星期五上午10:15触发
* “0 15 10 ? \* 6L 2002-2005” 2002年至2005年的每月的最后一个星期五上午10:15触发
* “0 15 10 ? \* 6#3” 每月的第三个星期五上午10:15触发

有些子表达式能包含一些范围或列表

例如：子表达式（天（星期））可以为 “MON-FRI”，“MON，WED，FRI”，“MON-WED,SAT”

“\*”字符代表所有可能的值   
“/”字符用来指定数值的增量

例如：在子表达式（分钟）里的“0/15”表示从第0分钟开始，每15分钟   
在子表达式（分钟）里的“3/20”表示从第3分钟开始，每20分钟（它和“3，23，43”）的含义一样

“？”字符仅被用于天（月）和天（星期）两个子表达式，表示不指定值   
当2个子表达式其中之一被指定了值以后，为了避免冲突，需要将另一个子表达式的值设为“？”

“L” 字符仅被用于天（月）和天（星期）两个子表达式，它是单词“last”的缩写   
如果在“L”前有具体的内容，它就具有其他的含义了。

例如：“6L”表示这个月的倒数第６天   
注意：在使用“L”参数时，不要指定列表或范围，因为这会导致问题

W 字符代表着平日(Mon-Fri)，并且仅能用于日域中。它用来指定离指定日的最近的一个平日。大部分的商业处理都是基于工作周的，所以 W 字符可能是非常重要的。

例如，日域中的 15W 意味着 “离该月15号的最近一个平日。” 假如15号是星期六，那么 trigger 会在14号(星期五)触发，因为星期四比星期一离15号更近。

C：代表“Calendar”的意思。它的意思是计划所关联的日期，如果日期没有被关联，则相当于日历中所有日期。

例如5C在日期字段中就相当于日历5日以后的第一天。1C在星期字段中相当于星期日后的第一天。

| **字段** | **允许值** | **允许的特殊字符** |
| --- | --- | --- |
| 秒 | 0~59 | , - \* / |
| 分 | 0~59 | , - \* / |
| 小时 | 0~23 | , - \* / |
| 日期 | 1-31 | , - \* ? / L W C |
| 月份 | 1~12或者JAN~DEC | , - \* / |
| 星期 | 1~7或者SUN~SAT | , - \* ? / L C # |
| 年（可选） | 留空，1970~2099 | , - \* / |