SpringTask是Spring自主研发的轻量级定时任务工具,相比于Quartz更加简单方便，且不需要引入其他依赖即可使用。

#### Cron表达式

一个字符串，包括6-7个时间元素，在SpringTask中可以用于指定任务的执行时间。

Seconds Minutes Hours DayofMonth Month DayofWeek

| **时间元素** | **可出现的字符** | **有效数值范围** |
| --- | --- | --- |
| Seconds | , - \* / | 0-59 |
| Minutes | , - \* / | 0-59 |
| Hours | , - \* / | 0-23 |
| DayofMonth | , - \* / ? L W | 0-31 |
| Month | , - \* / | 1-12 |
| DayofWeek | , - \* / ? L # | 1-7或SUN-SAT |

特殊字符说明：

| **字符** | **作用** | **举例** |
| --- | --- | --- |
| , | 列出枚举值 | 在Minutes域使用5,10，表示在5分和10分各触发一次 |
| - | 表示触发范围 | 在Minutes域使用5-10，表示从5分到10分钟每分钟触发一次 |
| \* | 匹配任意值 | 在Minutes域使用\*, 表示每分钟都会触发一次 |
| / | 起始时间开始触发，每隔固定时间触发一次 | 在Minutes域使用5/10,表示5分时触发一次，每10分钟再触发一次 |
| ? | 在DayofMonth和DayofWeek中，用于匹配任意值 | 在DayofMonth域使用?,表示每天都触发一次 |
| # | 在DayofMonth中，确定第几个星期几 | 1#3表示第三个星期日 |
| L | 表示最后 | 在DayofWeek中使用5L,表示在最后一个星期四触发 |
| W | 表示有效工作日(周一到周五) | 在DayofMonth使用5W，如果5日是星期六，则将在最近的工作日4日触发一次 |

#### 业务场景说明

* 用户对某商品进行下单操作；
* 系统需要根据用户购买的商品信息生成订单并锁定商品的库存；
* 系统设置了60分钟用户不付款就会取消订单；
* 开启一个定时任务，每隔10分钟检查下，如果有超时还未付款的订单，就取消订单并取消锁定的商品库存。

#### 添加SpringTask的配置

只需要在配置类中添加一个@EnableScheduling注解即可开启SpringTask的定时任务能力。

@Configuration

@EnableScheduling  
public class SpringTaskConfig {  
  
}

添加OrderTimeOutCancelTask来执行定时任务：

*/\*\**

*\* <p>  
 \* 订单超时取消并解锁库存的定时器  
 \* <p>  
 \*  
 \** ***@author*** *chance  
 \** ***@since*** *2019-12-31  
 \*/*

@Componentpublic class OrderTimeOutCancelTask {  
 private Logger LOGGER = LoggerFactory.*getLogger*(OrderTimeOutCancelTask.class);

*/\*\*  
 \* cron表达式：Seconds Minutes Hours DayofMonth Month DayofWeek [Year]  
 \* 每10分钟扫描一次，扫描设定超时时间之前下的订单，如果没支付则取消该订单  
 \*/*  
 @Scheduled(cron = "0 0/10 \* ? \* ?")  
 private void cancelTimeOutOrder() {  
 // *TODO:2019/5/3此处应调用取消订单的方法，具体查看mall项目源码* LOGGER.info("取消订单，并根据sku编号释放锁定库存");  
 }  
}