



链滴

# SpringTask

作者: [Secret](#)

原文链接: <https://ld246.com/article/1650032558091>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# SpringTask

## 一、定时任务

定时任务的应用场景

- 月底会员扣费
- 会员到期
- 生日祝福
- 等等

## 二、@Scheduled()参数

定义在方法上

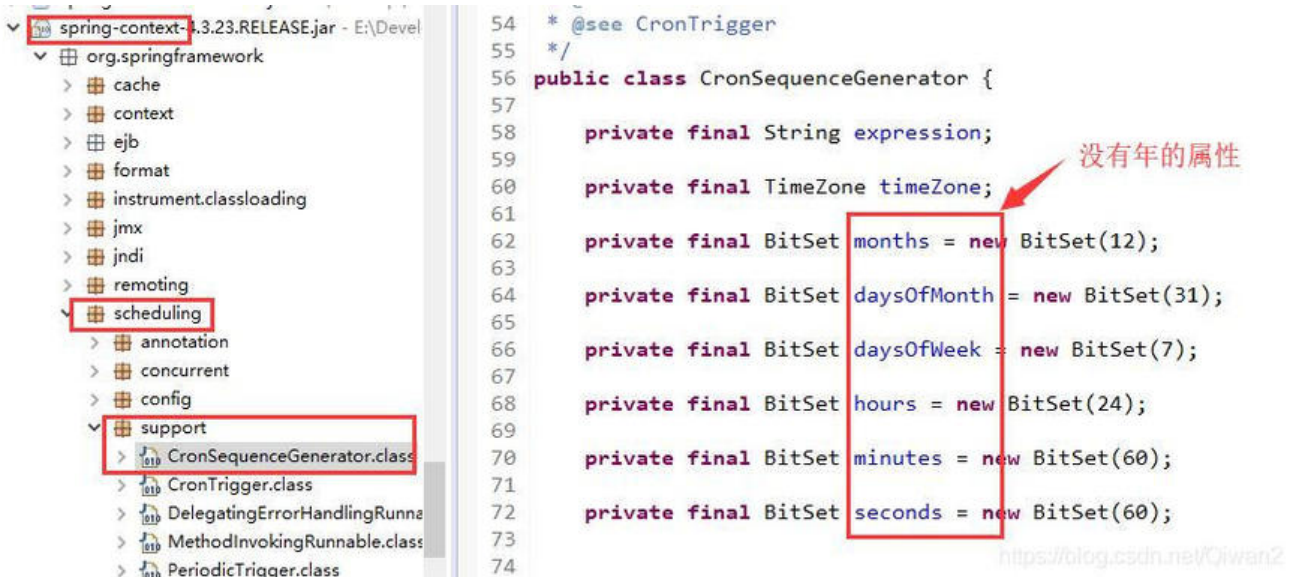
```
public @interface Scheduled {
    String cron() default ""; //表达式, 指定任务在特定时间执行;
    String zone() default ""; //时区, 默认为当前时区。
    long fixedDelay() default -1; //表示上一次任务执行完成后多久再次执行, 参数类型为long, 单位ms;
    String fixedDelayString() default ""; //与fixeddelay含义一样, 只是参数类型变为string;
    long fixedRate() default -1; //表示按一定的频率执行任务, 即每次开始执行的时间间隔致, 参数类型为long, 单位ms;
    String fixedRateString() default ""; //与fixedrate的含义一样, 只是将参数类型变为string;
    long initialDelay() default -1; //表示延迟多久再第一次执行任务, 参数类型为long, 单位s;
    String initialDelayString() default ""; //与initialdelay的含义一样, 只是将参数类型变为string

    TimeUnit timeUnit() default TimeUnit.MILLISECONDS;
}
```

### 2.1 cron

- [秒] [分] [时] [日] [月] [周]
- spring自带的定时任务cron表达式不支持年

如图:



上图类中有如下源代码：

```
private void parse(string expression) throws illegalargumentexception {
    string[] fields = stringutils.tokenizetostringarray(expression, " ");
    if (!arevalidcronfields(fields)) {
        throw new illegalargumentexception(string.format"cron expression must consist of 6 fields (found %d in \"%s\")", fields.length, expression);
    }
    doparse(fields);
}
private static boolean arevalidcronfields(string[] fields) {
    return (fields != null && fields.length == 6);
}
}
```

代码中expression参数即cron表达式，所以在spring自带的定时任务中正确的cron表达式只能是6个，否则会抛出illegalargumentexception提示：cron expression must consist of 6 fields（cron表达式必须由6个域组成）。

域 配符	必填	允许值	允许的
秒 (seconds) - * /	是	0-59整数	
分 (minutes) - * /	是	0-59整数	
时 (hours) - * /	是	0-23整数	
日 (daysofmonth) 天数)	, - * ? /   w	是	1-31整数（需要考虑月
月 (months) - * /	是	1-12整数 或 jan-dec	
周 (daysofweek) - * ? /   #	是	1-7整数 或 sun-sat	

## 通配符说明

- **\***: 表示匹配该域的任意值。在minutes域使用 \* 表示每分钟。在months里表示每个月。在daysofweek域表示一周的每一天。
- **?**: 只能用在daysofmonth和daysofweek两个域, 表示不指定值, 当两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?。因为daysofmonth和daysofweek会互影响。例如想在每月的2号触发调度, 不管2号是周几, 则只能使用如下写法: 0 0 0 2 \* ?, 其中第一位只能用?, 而不能使用\*, 如果使用\*表示不管周几都会触发。
- **-**: 表示范围。例如在minutes域使用5-20, 表示从5分到20分钟每分钟触发一次
- **/**: 表示起始时间开始触发, 然后每隔固定时间触发一次。例如在minutes域使用5/20, 则意味着当前小时的第5分钟开每20分钟触发一次。
- **,**: 表示列出枚举值。例如: 在minutes域使用5,20, 则意味着在5分和20分时各触发一次。
- **l**: 表示最后, 是单词“last”的缩写, 只能出现在daysofweek和dayofmonth域。在daysofweek使用5l意思是在指定月的最后的一个星期四触发。在dayofmonth域使用5l或者fril意思是在指定月的数第5天触发。在使用l参数时, 不要指定列表或范围。
- **w**: 表示有效工作日(周一到周五), 只能出现在daysofmonth域, 系统将在离指定日期的最近的有效工作日触发事件。例如: 在daysofmonth使用5w, 如果5号是周六, 则将在最近的工作日周五, 即4号发。如果5号是周日, 则在6日(周一)触发。如果5日在星期一到星期五中的一天, 则就在5日触发。另, w的最近寻找不会跨过月份。
- **lw**: 这两个字符可以连用, 表示指定月的最后一个工作日。
- **#**: 用于确定每个月第几个周几, 只能出现在daysofmonth域。例如在4#2, 表示某月的第二个周。

## 常用表达式示例

- **0/2 \* \* \* \* ?** 表示每2秒 执行任务
- **0 0/2 \* \* \* ?** 表示每2分钟 执行任务
- **0 0 2 1 \* ?** 表示在每月的1日的凌晨2点调整任务
- **0 15 10 ? \* mon-fri** 表示周一到周五每天上午10:15执行作业
- **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 \* 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-wed,sat** 周一至周三和周六的上午10:15触发
- **0 15 10 15 \* ?** 每月15日上午10:15触发

- 0 15 10 | \* ? 每月最后一日的上午10:15触发
- 0 15 10 ? \* 6| 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

例

```
@Scheduled("*/6 * * * * ?")
public void test(){
    System.out.println("每隔六秒执行一次-->" + new Date());
}
```

生成器工具地址-<http://cron.qqe2.com/>

## 2.2 fixedDelay

例

```
@Scheduled(fixedDelay = 1000 * 10)
public void test(){
    System.out.println("每隔10秒执行一次-->" + new Date());
}
```

## 2.3 fixedRate

例

```
@Scheduled(fixedRate = 1000 * 10)
public void test(){
    System.out.println("每隔10秒执行一次-->" + new Date());
}
```

## 2.4 initialDelay

例

```
@Scheduled(initialDelay = 1000, fixedRate = 3000)
public void test(){
    System.out.println("第一次延迟1s, 之后按照fixedRate的规则执行--->" + new Date());
}
```

## 2.5 注意事项

- 1、定时器的参数有两种写法是用cron表达式，或者使用fixeddelay、fixedrate等参数直接配置  
需要注意的是 使用cron表达式的方法，在项目首次启动后**不会直接运行**，而是等到执行周期才会执行  
而使用第二种方式的定时器方法，在项目启动成功后会**马上开始执行一次**，再按照时间周期执行
- 2、定时器默认为单线程，所以如果项目中使用多个定时器要配置线程池

## 三、实战



```
gCacheTest\target\classes started by admin in E:\git项目\java普通项目\SpringCacheTest)
2022-04-15 21:16:26 [main] INFO com.springcache.DemoApplication - No active profile set, falling
back to 1 default profile: "default"
2022-04-15 21:16:27 [main] INFO o.s.d.r.config.RepositoryConfigurationDelegate - Multiple
pring Data modules found, entering strict repository configuration mode!
2022-04-15 21:16:27 [main] INFO o.s.d.r.config.RepositoryConfigurationDelegate - Bootstrap
ing Spring Data JDBC repositories in DEFAULT mode.
2022-04-15 21:16:27 [main] INFO o.s.d.r.config.RepositoryConfigurationDelegate - Finished S
ring Data repository scanning in 11 ms. Found 0 JDBC repository interfaces.
2022-04-15 21:16:27 [main] INFO o.s.d.r.config.RepositoryConfigurationDelegate - Multiple
pring Data modules found, entering strict repository configuration mode!
2022-04-15 21:16:27 [main] INFO o.s.d.r.config.RepositoryConfigurationDelegate - Bootstrap
ing Spring Data Redis repositories in DEFAULT mode.
2022-04-15 21:16:27 [main] INFO o.s.d.r.config.RepositoryConfigurationDelegate - Finished S
ring Data repository scanning in 5 ms. Found 0 Redis repository interfaces.
2022-04-15 21:16:28 [main] INFO o.s.boot.web.embedded.tomcat.TomcatWebServer - Tomcat
initialized with port(s): 8080 (http)
2022-04-15 21:16:28 [main] INFO org.apache.catalina.core.AprLifecycleListener - An older ver
sion [1.2.26] of the Apache Tomcat Native library is installed, while Tomcat recommends a min
imum version of [1.2.30]
2022-04-15 21:16:28 [main] INFO org.apache.catalina.core.AprLifecycleListener - Loaded Apa
che Tomcat Native library [1.2.26] using APR version [1.7.0].
2022-04-15 21:16:28 [main] INFO org.apache.catalina.core.AprLifecycleListener - APR capabili
ties: IPv6 [true], sendfile [true], accept filters [false], random [true], UDS [true].
2022-04-15 21:16:28 [main] INFO org.apache.catalina.core.AprLifecycleListener - APR/OpenS
SL configuration: useAprConnector [false], useOpenSSL [true]
2022-04-15 21:16:28 [main] INFO org.apache.catalina.core.AprLifecycleListener - OpenSSL su
ccessfully initialized [OpenSSL 1.1.1i 8 Dec 2020]
2022-04-15 21:16:28 [main] INFO org.apache.catalina.core.StandardService - Starting service
Tomcat]
2022-04-15 21:16:28 [main] INFO org.apache.catalina.core.StandardEngine - Starting Servlet
engine: [Apache Tomcat/9.0.58]
2022-04-15 21:16:28 [main] INFO o.a.c.core.ContainerBase.[Tomcat].[localhost].[/] - Initializin
Spring embedded WebApplicationContext
2022-04-15 21:16:28 [main] INFO o.s.b.w.s.c.ServletWebServerApplicationContext - Root We
bApplicationContext: initialization completed in 1695 ms
2022-04-15 21:16:28 [main] INFO com.springcache.Config.CacheConfig - 初始化 -> [Redis Ca
cheErrorHandler]
2022-04-15 21:16:29 [main] INFO com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Starting
..
2022-04-15 21:16:30 [main] INFO com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Start c
ompleted.
2022-04-15 21:16:30 [main] INFO o.s.boot.web.embedded.tomcat.TomcatWebServer - Tomcat
started on port(s): 8080 (http) with context path ""
定时项目测试1---date-->Fri Apr 15 21:16:30 CST 2022
定时项目测试2---date-->Fri Apr 15 21:16:30 CST 2022
2022-04-15 21:16:30 [main] INFO com.springcache.DemoApplication - Started DemoApplicat
ion in 4.368 seconds (JVM running for 6.444)
定时项目测试1---date-->Fri Apr 15 21:16:31 CST 2022
定时项目测试2---date-->Fri Apr 15 21:16:31 CST 2022
定时项目测试1---date-->Fri Apr 15 21:16:32 CST 2022
定时项目测试2---date-->Fri Apr 15 21:16:32 CST 2022
定时项目测试1---date-->Fri Apr 15 21:16:33 CST 2022
定时项目测试2---date-->Fri Apr 15 21:16:33 CST 2022
```

可以看到，只要项目启动，他就会一直执行

`<span style="color:red">文章参考</span>`

`<a href = "https://www.10qianwan.com/articledetail/836813.html">参考文章1</a>`

`<a href="https://blog.csdn.net/weixin_44356673/article/details/105827041">参考文章2</a>`