

java.util下的Timer, TimerTask和quartz的比较(2) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_javautil_E4_c104_144585.htm 这里面我们可以看到，当我们定义了任务执行 QuartzReport 类后，需要定一个Scheduler类用来执行计划任务。一个JobDetail 类来描述这个任务的信息，包括任务信息，任务所在组，任务执行的类。然后还要定义一个触发器，类似的也包括触发器名，触发器所在组，触发器触发时间设定。最后是调度器Scheduler类执行计划任务。基本上一个计划任务执行的流程就完成了。当然，我们还看到了上面红色代表的内容，这些内容主要是提供在job方法执行的时候所需要的参数的提供。这里使用了JobDataMap 类，它其实就是实现了map的特殊应用的一个类，使用方法与Map很相似。我们可以用 put () 输入参数。在Job类中使用cntxt.getJobDetail ()。getJobDataMap ()。get ("type") 方法获取输入的参数的值。这里的cntxt 是 JobExecutionContext .就是包含任务执行上下文的一个信息类。这样我们的一个基本的任务执行就可以搞定了。触发器有两类：SimpleTrigger andCronTrigger. .SimpleTrigger主要提供了跟 java.util.Timer 类相似的功能.....你可以在里面定义任务的起始时间，终止时间，任务的执行次数，任务执行的中间间隔 . 而 CronTrigger类主要提供了更高级的任务调度时间设置，例如 每个星期天的早上7点 . CronTrigger的时间设置说明在后来介绍。下面我们介绍一下在 J2EE 环境下如何来使用 Quartz . 首先，我们要配置 web.xml ，添加一下内容，主要是Quartz 的初始化， QuartzInitializer Quartz Initializer Servlet

org.quartz.ee.servlet.QuartzInitializerServlet 1 然后还要有一个Quartz 的配置文件 quartz.properties 放置在 WEB-INF/classes 目录下面。StdScheduleFactory () 会读取它。配置如下##
Configure Main Scheduler Properties #
org.quartz.scheduler.instanceName = TestScheduler
org.quartz.scheduler.instanceId = one ## Configure ThreadPool #
org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool
org.quartz.threadPool.threadCount = 5
org.quartz.threadPool.threadPriority = 4 # 100Test 下载频道开通
，各类考试题目直接下载。详细请访问 www.100test.com