

详细讲解Quartz如何从入门到精通 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022__E8_AF_A6_E7_BB_86_E8_AE_B2_E8_c104_645110.htm

Quartz是一个开源的作业调度框架，它完全由Java写成，并设计用于J2SE和J2EE应用中。它提供了巨大的灵活性而不牺牲简单性。你能够用它来为执行一个作业而创建简单的或复杂的调度。它有很多特征，如：数据库支持，集群，插件，EJB作业预构建，JavaMail及其它，支持cron-like表达式等等。你曾经需要应用执行一个任务吗？这个任务每天或每周星期二晚上11：30，或许仅仅每个月的最后一天执行。一个自动执行而无须干预的任务在执行过程中如果发生一个严重错误，应用能够知到其执行失败并尝试重新执行吗？你和你的团队是用Java编程吗？如果这些问题中任何一个你回答是，那么你应该使用Quartz调度器。旁注：Matrix目前就大量使用到了Quartz。比如，排名统计功能的实现，在Jmatrix里通过Quartz定义了一个定时调度作业，在每天凌晨一点，作业开始工作，重新统计大家的Karma和排名等。还有，RSS文件的生成，也是通过Quartz定义作业，每隔半个小时生成一次RSS XML文件。Quartz让作业调度简单 Quartz是一个完全由Java编写的开源作业调度框架。不要让作业调度这个术语吓着你。尽管Quartz框架整合了许多额外功能，但就其简易形式看，你会发现它易用得简直让人受不了！。简单地创建一个实现org.quartz.Job接口的Java类。Job接口包含唯一的方法：在你的Job接口实现类里面，添加一些逻辑到execute()方法。一旦你配置好Job实现类并设定好调度时间表，Quartz将密切注意剩余时间。当

调度程序确定该是通知你的作业的时候，Quartz框架将调用你Job实现类（作业类）上的execute()方法并允许做它该做的事情。无需报告任何东西给调度器或调用任何特定的东西。仅仅执行任务和结束任务即可。如果配置你的作业在随后再次被调用，Quartz框架将在恰当的时间再次调用它。如果你使用了其它流行的开源框架象struts，你会对Quartz的设计和部件感到舒适。虽然两个开源工程是解决完全不同的问题，还是有很多相似的之处，就是开源软件用户每天感觉很舒适。Quartz能用在单机J2SE应用中，作为一个RMI服务器，也可以用在web应用中，甚至也可以用在J2EE应用服务器中。

Quartz的发展史

尽管Quartz今年开始受到人们注意，但还是暂时流行。Quartz由James House创建并最初于2001年春天被加入sourceforge工程。接下来的几年里，有许多新特征和版本出现，但是直到项目迁移到新的站点并成为 OpenSymphony项目家族的一员，才开始真正启动并受到应有的关注。James House仍然和几个协助他的业余开发者参与大量开发工作。Quartz开发团队今年能发布几个新版本，包括当前正处在候选发布阶段的1.5版。

上手Quartz

Quartz工程驻留在OpenSymphony站点上。在Quartz站点上可以找到许多有用的资源：JavaDocs，包含指南的文档，CVS访问，用户和开发者论坛的连接，当然也有下载。从下载连接取得Quartz的发布版本，并且解压到本地目录。这个下载文件包含了一个预先构建好的Quartz二进制文件（quartz.jar），你可以将它放进自己的应用中。Quartz框架只需要少数的第三方库，并且这些三方库是必需的，你很可能已经在使用这些库了。你要把Quartz的安装目录的/lib/core 和 /lib/optional目录中的第三方

库加进你自己的工程中。大多数第三方库是我们所熟知和喜欢的标准Jakarta Commons库，像Commons Logging, Commons BeantUtils等等。 quartz.properties文件 Quartz有一个叫做quartz.properties的配置文件，它允许你修改框架运行时环境。缺省是使用Quartz.jar里面的 quartz.properties文件。当然，你应该创建一个quartz.properties文件的副本并且把它放入你工程的classes目录中以便类装载机找到它。一旦将Quartz.jar文件和第三方库加到自己的工程里面并且quartz.properties文件在工程的classes目录中，就可以创建作业了。然而，在做这之前，我们暂且回避一下先简短讨论一下Quartz架构。 Quartz内部架构在规模方面， Quartz跟大多数开源框架类似。大约有300个Java类和接口，并被组织到12个包中。这可以和Apache Struts把大约325个类和接口以及组织到11个包中相比。尽管规模几乎不会用来作为衡量框架质量的一个特性，但这里的关键是quartz内含很多功能，这些功能和特性集是否成为、或者应该成为评判一个开源或非开源框架质量的因素。 Quartz调度器 Quartz框架的核心是调度器。调度器负责管理Quartz应用运行时环境。调度器不是靠自己做所有的工作，而是依赖框架内一些非常重要的部件。 Quartz不仅仅是线程和线程管理。为确保可伸缩性， Quartz采用了基于多线程的架构。启动时，框架初始化一套worker线程，这套线程被调度器用来执行预定的作业。这就是Quartz怎样能并发运行多个作业的原理。 Quartz依赖一套松耦合的线程池管理部件来管理线程环境。本文中，我们会多次提到线程池管理，但Quartz里面的每个对象是可配置的或者是可定制的。所以，例如，如果你想要插进自己线程池管理设施，我猜你一定

能！作业用Quartz的行话讲，作业是一个执行任务的简单Java类。任务可以是任何Java代码。只需你实现org.quartz.Job接口并且在出现严重错误情况下抛出JobExecutionException异常即可。Job接口包含唯一的一个方法execute()，作业从这里开始执行。一旦实现了Job接口和execute()方法，当Quartz确定该是作业运行的时候，它将调用你的作业。Execute()方法内就完全是你要做的事情。下面有一些你要在作业里面做事情的例子：用JavaMail（或者用其他的像Commons Net一样的邮件框架）发送邮件 创建远程接口并且调用在EJB上的方法 获取Hibernate Session，查询和更新关系数据库里的数据 使用OSWorkflow并且从作业调用一个工作流 使用FTP和到处移动文件 调用Ant构建脚本开始预定构建 这种可能性是无穷的，正是这种无限可能性使得框架功能如此强大。Quartz给你提供了一个机制来建立具有不同粒度的、可重复的调度表，于是，你只需创建一个Java类，这个类被调用而执行任务。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