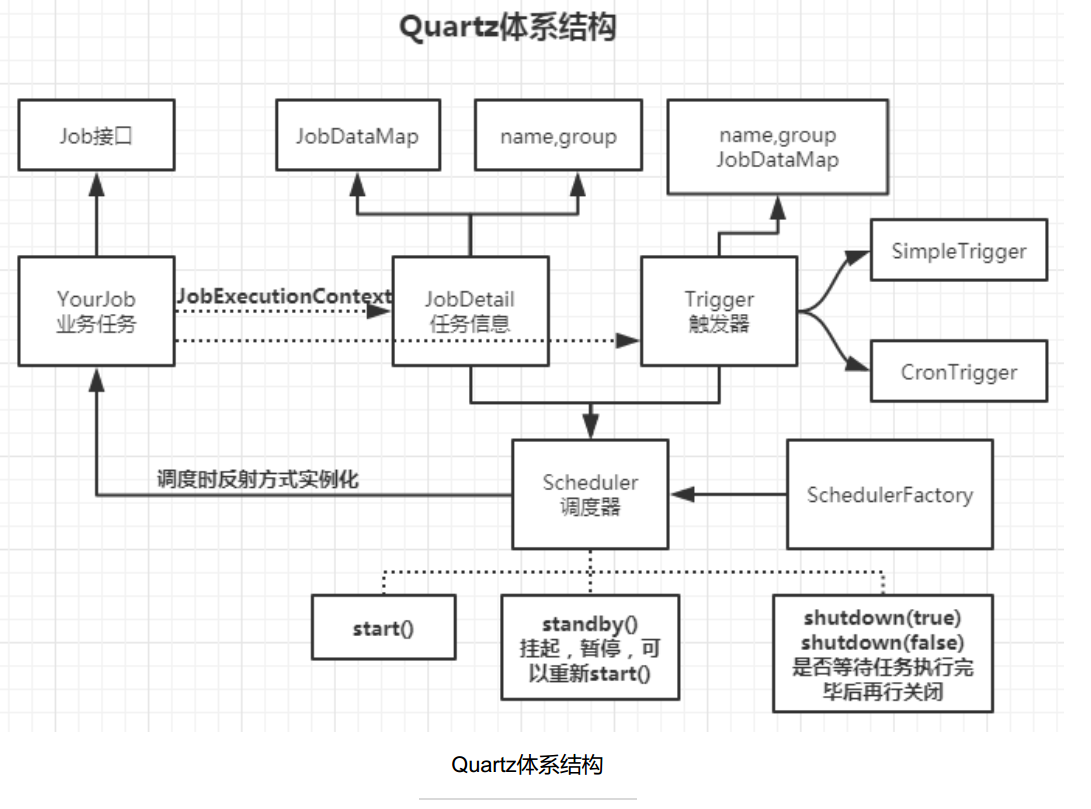
# Quartz是什么？

Quartz是OpenSymphony开源组织在Job scheduling领域又一个开源项目--作业调度框架，它可以与J2EE与J2SE应用程序相结合也可以单独使用。Quartz可以用来创建简单或为运行十个，百个，甚至是好几万个Jobs这样复杂的程序。Jobs可以做成标准的Java组件或 EJBs。Quartz的最新版本为Quartz 2.3.0。--百度百科(2017年11月6日11:03:12)

# 2.Quartz的系统架构



# 3.Quartz怎么用？

Quartz有三大核心：调度器、Job、触发器。

使用步骤如下：

1. 使用SchedulerFactory创建Scheduler调度器实例
2. 使用JobBuilder创建JobDetail实例（需要创建一个类实现job接口，该类的逻辑就是调度做的事情。）
3. 定义触发规则，触发规则有两种，一种是java代码的方式，一种是cron表达式的方式。推荐使用cron表达式，更加强大。Cron表达式的生成可以到：<http://www.pppet.net/>
4. 调用scheduler对象的start()方法开始调度。

代码如下：

|  |
| --- |
| **package** com.keji.blog.codelibrary.scheduling.quartz;  **import** org.quartz.CronScheduleBuilder; **import** org.quartz.JobBuilder; **import** org.quartz.JobDetail; **import** org.quartz.Scheduler; **import** org.quartz.SchedulerException; **import** org.quartz.SimpleScheduleBuilder; **import** org.quartz.SimpleTrigger; **import** org.quartz.TriggerBuilder; **import** org.quartz.impl.StdSchedulerFactory;  */\*\*  \** ***@author*** *wb-ny291824  \** ***@version*** *$Id: ScheduleTest.java, v 0.1 2017-11-02 16:46 wb-ny291824 Exp $$  \*/* **public class** ScheduleTest {  **public static void** main(String[] args) {   StdSchedulerFactory stdSchedulerFactory = **new** StdSchedulerFactory();  Scheduler scheduler = **null**;  **try** {  *//通过schedulerFactory获取一个调度器* scheduler = stdSchedulerFactory.getScheduler();  } **catch** (Exception e) {  System.***out***.println(**"创建调度器失败"**);  e.printStackTrace();  }   *// 创建jobDetail实例，绑定Job实现类  // 指明job的名称，所在组的名称，以及绑定job类* JobDetail jobDetail = JobBuilder.*newJob*(HelloQuartzJob.**class**).withIdentity(**"JobName"**, **"JobGroupName"**).build();   *//定义调度触发规则  //1.SimpleTrigger的方式(每3秒执行一次,总共执行6+1次)* SimpleTrigger trigger = TriggerBuilder.*newTrigger*().withIdentity(**"SimpleTrigger"**, **"SimpleTriggerGroup"**)  .withSchedule(SimpleScheduleBuilder.*repeatSecondlyForever*(3).withRepeatCount(6)).startNow().build();  *//2.corn表达式的方式(每5秒执行一次)* TriggerBuilder.*newTrigger*().withIdentity(**"CronTrigger1"**, **"CronTriggerGroup"**).withSchedule(  CronScheduleBuilder.*cronSchedule*(**"\*/5 \* \* \* \* ?"**)).startNow().build();   **try** {  *// 把作业和触发器注册到任务调度中* **if** (**null** != scheduler) {  scheduler.scheduleJob(jobDetail, trigger);  }  *//启动调度* scheduler.start();  *// 停止调度  //scheduler.shutdown();* } **catch** (SchedulerException e) {  e.printStackTrace();  }   } } |

|  |
| --- |
| **package** com.keji.blog.codelibrary.scheduling.quartz;  **import** org.quartz.Job; **import** org.quartz.JobExecutionContext; **import** org.quartz.JobExecutionException;  */\*\*  \** ***@author*** *wb-ny291824  \** ***@version*** *$Id: HelloQuartzJob.java, v 0.1 2017-11-02 16:45 wb-ny291824 Exp $$  \*/* **public class** HelloQuartzJob **implements** Job{  @Override  **public void** execute(JobExecutionContext context) **throws** JobExecutionException {  System.***out***.println(**"Hello Quartz!"**);  } } |