Quartz Glass

A Simple Web UI for Quartz
Bingoo Huang

Spring XML Job

```xml
<bean id="MyJob每30分钟" class="org.n3r.quartz.glass.job.GlassTriggerFactoryBean">
    <property name="jobClass" value="org.n3r.quartz.glass.job.demo.MyJob"/>
    <property name="scheduler" value="Every 30 minutes"/>
    <property name="triggerDataMap" value="staticType=hotSale"/>
</bean>

<bean id="MyJob每小时开始20分钟" class="org.n3r.quartz.glass.job.GlassTriggerFactoryBean">
    <property name="jobClass" value="org.n3r.quartz.glass.job.demo.MyJob"/>
    <property name="scheduler" value="0 20 * * * ?"/>
    <property name="triggerDataMap" value="staticType=商品静态"/>
</bean>

<bean id="MyJob每天凌晨3点" class="org.n3r.quartz.glass.job.GlassTriggerFactoryBean">
    <property name="jobClass" value="org.n3r.quartz.glass.job.demo.MyJob"/>
    <property name="scheduler" value="At 03:00"/>
    <property name="jobDataMap" value="staticType=首页静态"/>
</bean>

<bean id="POJO每小时40分钟干活" class="org.n3r.quartz.glass.job.GlassTriggerFactoryBean">
    <property name="jobClass" value="org.n3r.quartz.glass.job.demo.PojoJob"/>
    <property name="scheduler" value="At ??:40"/>
    <property name="triggerDataMap" value="orderType=退货订单"/>
</bean>
```
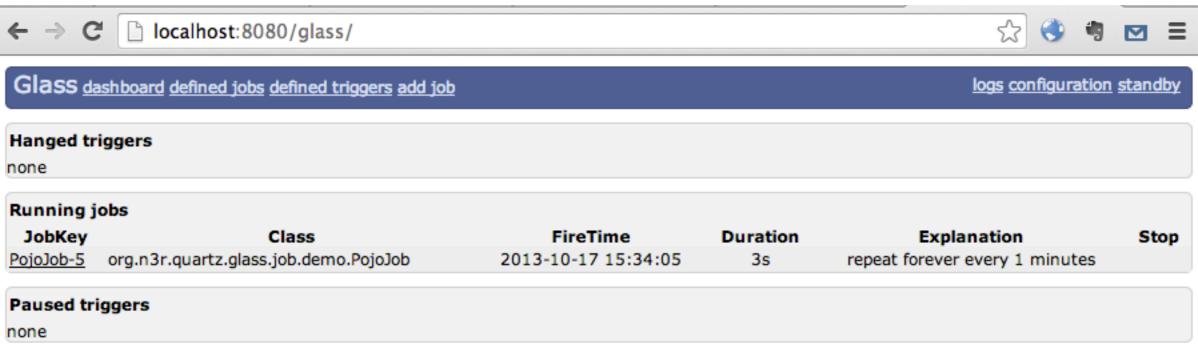
Glass UI - Triggers

localhost:8080/glass/triggers

Glass dashboard defined jobs defined triggers add job   logs configuration standby

Defined triggers

#	Type	JobKey	TriggerKey	Glass Scheduler	Start time	End time	Last fire	Next fire	Explanation	Pause / Resume
1	cron	MyJob-3	MyJob每天凌晨3点	At 03:00	2013-10-17 15:29:05			2013-10-18 03:00:00	At 3:00:00 AM (0 0 3 ? * *)	❚❚
2	simple	MyJob-1	MyJob每30分钟	Every 30 minutes	2013-10-17 15:29:04 running		2013-10-17 15:29:04	2013-10-17 15:59:04	repeat forever every 30 minutes	❚❚
staticType=hotSale										
3	cron	MyJob-1	MyJob每小时开始20分钟	0 20 * * * ?	2013-10-17 15:29:05			2013-10-17 16:20:00	At 20 minutes past the hour (0 20 * * * ?)	❚❚
staticType=商品静态										
4	simple	PojoJob-5	POJO每1分钟干活	Every 1 minute	2013-10-17 15:29:05 running		2013-10-17 15:29:05	2013-10-17 15:30:05	repeat forever every 1 minutes	❚❚
logLevel=INFO, orderType=退货订单										
5	cron	PojoJob-4	POJO每小时40分钟干活	At ??:40	2013-10-17 15:29:05			2013-10-17 15:40:00	At 40 minutes past the hour (0 40 * * * ?)	❚❚
orderType=退货订单										

version 0.1.0 (Mon Oct 14 12:24:36 CST 2013)

Glass UI - Jobs

localhost:8080/glass/jobs

Glass dashboard defined jobs defined triggers add job | logs configuration standby

Defined jobs

#	Jobkey	Class	Desc	Team	Created	JobDataMap
1	MyJob-3	org.n3r.quartz.glass.job.demo.MyJob	静态化演示任务	火箭队	2013-10-16	staticType=首页静态
2	MyJob-1	org.n3r.quartz.glass.job.demo.MyJob	静态化演示任务	火箭队	2013-10-16	
3	PojoJob-5	org.n3r.quartz.glass.job.demo.PojoJob	淘宝订单同步	小牛队	2013-10-16	
4	PojoJob-4	org.n3r.quartz.glass.job.demo.PojoJob	淘宝订单同步	小牛队	2013-10-16	

version 0.1.0 (Mon Oct 14 12:24:36 CST 2013)

Glass UI - Dashboard

Glass UI - Job

Job PojoJob-5
edit logs delete

Class	org.n3r.quartz.glass.job.demo.PojoJob
Desc	淘宝订单同步
Team	小牛队
Created	2013-10-16
Disallow concurrent execution	true
Persist job data after execution	true
Data map	[ ] [Fire Now]

Jobs arguments

Name	Description	Sample Values
logLevel	Log level used for this job.	[DEBUG, INFO, WARN, ERROR]

Defined triggers
add cron add simple

Type	TriggerKey	Glass Scheduler	Start time	End time	Last fire time	Next fire time	Explanation	Pause / Resume	Delete
simple	POJO每1分钟干活	Every 1 minute	2013-10-17 15:29:05 running		2013-10-17 15:36:05	2013-10-17 15:37:05	repeat forever every 1 minutes	❙❙	🗑
logLevel=INFO, orderType=退货订单									

Recent executions

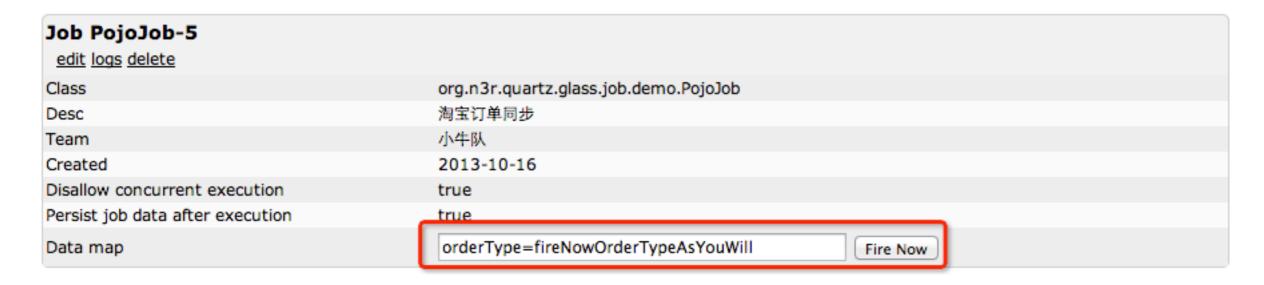| JobKey	TriggerKey	Start	End	Job class	DataMap
PojoJob-5	POJO每1分钟干活	2013-10-17 15:36:05	running	org.n3r.quartz.glass.job.demo.PojoJob	orderType=退货订单, logLevel=INFO
logs					
2013-10-17 15:36:15 INFO XXXXXXXX淘宝订单同步淘宝订单同步淘宝订单同步淘宝订单同步淘宝订单同步:退货订单					
PojoJob-5	POJO每1分钟干活	2013-10-17 15:35:05	2013-10-17 15:35:15	org.n3r.quartz.glass.job.demo.PojoJob	orderType=退货订单, logLevel=INFO
logs					

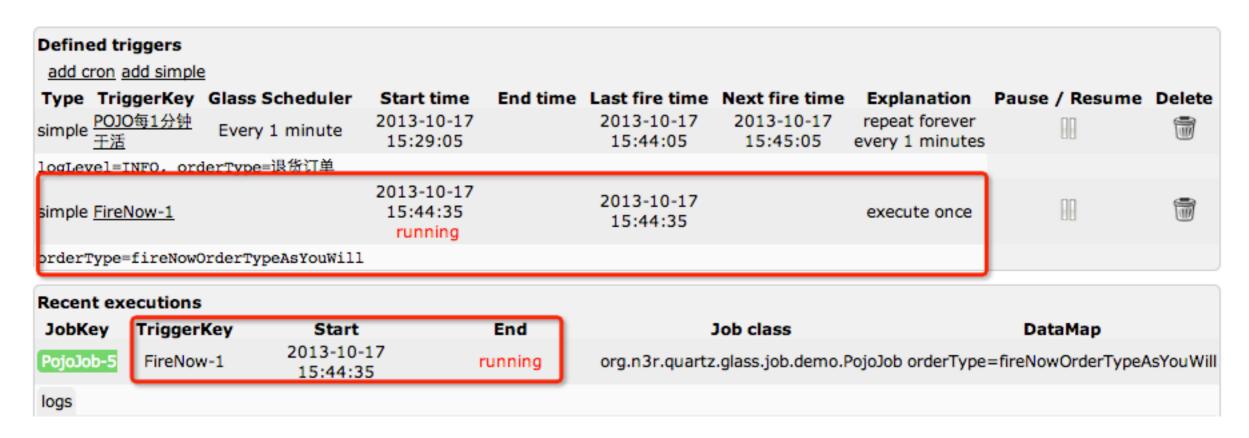view logs

How code Job - Quartz Way

```
01 @GlassJob(description = "静态化演示任务", team = "火箭队", created = "2013-10-16")
02 @PersistJobDataAfterExecution
03 @DisallowConcurrentExecution
04 public class MyJob implements Job {
05     @JobArgument(description = "静态化类型", sampleValues = {"hotSale", "discountSale"})
06     private String staticType;
07
08     @Override
09     public void execute(JobExecutionContext context) throws JobExecutionException {
10         try {
11             Thread.sleep(10000);
12         } catch (InterruptedException e) {
13             e.printStackTrace();
14         }
15         JobLogs.warn("静态化类型是: " + staticType);
16     }
17 }
```

How code Job - POJO Way

```
01 public class PojoJob {
02     private String orderType;
03
04     @GlassJob(description = "淘宝订单同步", team = "小牛队", created = "2013-10-16")
05     public void execute() {
06         try {
07             Thread.sleep(10000);
08         } catch (InterruptedException e) {
09             e.printStackTrace();
10         }
11         JobLogs.info("同步淘宝:" + orderType);
12     }
13
14     public String getOrderType() {
15         return orderType;
16     }
17
18     public void setOrderType(String orderType) {
19         this.orderType = orderType;
20     }
21 }
```

Fire Now

Job PojoJob-5

edit logs delete

Class	org.n3r.quartz.glass.job.demo.PojoJob
Desc	淘宝订单同步
Team	小牛队
Created	2013-10-16
Disallow concurrent execution	true
Persist job data after execution	true
Data map	orderType=fireNowOrderTypeAsYouWill [Fire Now]

Defined triggers

add cron add simple

Type	TriggerKey	Glass Scheduler	Start time	End time	Last fire time	Next fire time	Explanation	Pause / Resume	Delete
simple	POJO每1分钟干活	Every 1 minute	2013-10-17 15:29:05		2013-10-17 15:44:05	2013-10-17 15:45:05	repeat forever every 1 minutes	❚❚	🗑
logLevel=INFO, orderType=退货订单									
simple FireNow-1			2013-10-17 15:44:35 running		2013-10-17 15:44:35		execute once	❚❚	🗑
orderType=fireNowOrderTypeAsYouWill									

Recent executions

JobKey	TriggerKey	Start	End	Job class	DataMap
PojoJob-5	FireNow-1	2013-10-17 15:44:35	running	org.n3r.quartz.glass.job.demo.PojoJob	orderType=fireNowOrderTypeAsYouWill

logs

Simple Glass Scheduler

Glass Scheduler	Quartz Scheduler	Readable Explanation
Every 30 minutes more examples: Every 1 h/m/s	Simple interval=30m	repeat forever every 30 minutes
0 20 * * * ?	Cron	At 20 minutes past the hour
At 03:00	Cron 0 0 3 ? * *	At 3:00:00 AM (0 0 3 ? * *)
At ??:40	Cron 0 40 * * * ?	At 40 minutes past the hour (0 40 * * * ?)

Deployment-web.xml

```xml
01 <filter>
02     <filter-name>charsetFilter</filter-name>
03     <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
04     <init-param>
05         <param-name>encoding</param-name>
06         <param-value>UTF-8</param-value>
07     </init-param>
08 </filter>
09 <filter-mapping>
10     <filter-name>charsetFilter</filter-name>
11     <url-pattern>/*</url-pattern>
12 </filter-mapping>
13
14 <servlet>
15     <servlet-name>spring</servlet-name>
16     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
17     <init-param>
18         <param-name>contextConfigLocation</param-name>
19         <param-value>classpath:org/n3r/quartz/glass/springConfig.xml,
20             classpath*:spring*.xml</param-value>
21     </init-param>
22 </servlet>
23 <servlet-mapping>
24     <servlet-name>spring</servlet-name>
25     <url-pattern>/glass/*</url-pattern>
26 </servlet-mapping>
27
28 <welcome-file-list>
29     <welcome-file>glass/index</welcome-file>
30 </welcome-file-list>
```

TKS