

# Quartz Extension

# Table of Contents

How to configure/use .....	2
Classpath .....	2
Configure Quartz .....	2
Known issues.....	5
Dependencies .....	6

This module (`isis-module-quartz`) provides a wrapper around the [Quartz](#) scheduler.

# How to configure/use

You can either use this module "out-of-the-box", or you can fork this repo and extend to your own requirements.

## Classpath

Update your classpath by adding this dependency in your **webapp**'s project's **pom.xml**:

```
<dependency>
  <groupId>org.isisaddons.module.quartz</groupId>
  <artifactId>isis-module-quartz-dom</artifactId>
</dependency>
```

Check for later releases by searching [Maven Central Repo](<http://search.maven.org/#search|ga|1|isis-module-quartz-dom>).

For instructions on how to use the latest **-SNAPSHOT**, see the [contributors guide](#).

## Configure Quartz

There are many ways to configure Quartz; the instructions below are for a basic setup:

- First, add an entry in **web.xml**:

```
<servlet>
  <servlet-name>QuartzInitializer</servlet-name>
  <servlet-class>org.quartz.ee.servlet.QuartzInitializerServlet</servlet-class>
  <init-param>
    <param-name>config-file</param-name>
    <param-value>config/quartz.properties</param-value>
  </init-param>
  <init-param>
    <param-name>shutdown-on-unload</param-name>
    <param-value>true</param-value>
  </init-param>
  <init-param>
    <param-name>start-scheduler-on-load</param-name>
    <param-value>true</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

- Second, add a **config/quartz.properties** file:

```
org.quartz.scheduler.instanceName = SchedulerQuartzConfigXml
org.quartz.threadPool.threadCount = 1
org.quartz.jobStore.class = org.quartz.simpl.RAMJobStore
org.quartz.plugin.jobInitializer.class =
org.quartz.plugins.xml.XMLSchedulingDataProcessorPlugin
org.quartz.plugin.jobInitializer.fileNames = config/quartz-config.xml
org.quartz.plugin.jobInitializer.failOnFileNotFound = true
```

- Finally, add a `config/quartz-config.xml` file (also under in `src/main/resources` so is loaded from classpath):

```

<?xml version="1.0" encoding="UTF-8"?>
<job-scheduling-data
  xmlns="http://www.quartz-scheduler.org/xml/JobSchedulingData"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.quartz-scheduler.org/xml/JobSchedulingData
http://www.quartz-scheduler.org/xml/job_scheduling_data_1_8.xsd"
  version="1.8">

  <schedule>
    <job>
      <name>RunBackgroundCommandsJob</name>
      <group>CommandModule</group>
      <description>Runs all background commands</description>
      <job-
class>org.isisaddons.module.quartz.dom.jobs.RunBackgroundCommandsJob</job-class>
      <job-data-map>
        <entry>
          <key>user</key>
          <value>scheduler_user</value>
        </entry>
        <entry>
          <key>roles</key>
          <value>admin_role</value>
        </entry>
      </job-data-map>
    </job>

    <trigger>
      <cron>
        <name>RunBackgroundJobsEvery10Seconds</name>
        <job-name>RunBackgroundCommandsJob</job-name>
        <job-group>CommandModule</job-group>
        <cron-expression>0/10 * * * * ?</cron-expression>
      </cron>
    </trigger>

  </schedule>
</job-scheduling-data>

```

Details on the cron format can be found, for example, in [this tutorial](#). For example:

- `0 0/30 * * * ?` is every 30 minutes
- `0/10 * * * * ?` is every 10 seconds

# Known issues

None known at this time.

# Dependencies

Maven can report modules dependencies using:

```
mvn dependency:list -o -pl modules/ext/quartz/impl -D excludeTransitive=true
```

which, excluding the Incode Platform and Apache Isis modules, returns these compile/runtime dependencies:

```
org.quartz-scheduler:quartz:jar:2.2.1  
org.quartz-scheduler:quartz-jobs:jar:2.2.1
```

For further details on 3rd-party dependencies, see:

- [Quartz](#)