Spring provides excellent support for scheduling jobs based on [**cron expression**](http://en.wikipedia.org/wiki/Cron) also. This scheduling is possible with use of @Scheduled annotation. According to spring documentation:

Spring 3.0 also adds annotation support for both task scheduling and asynchronous method execution. The @Scheduled annotation can be added to a method along with trigger metadata.

In this post, I will show the means to use this feature in 4 different ways.

**Sections in this post:**

Explaining @Scheduled annotation

Task scheduling using fixed delay attribute in @Scheduled annotation

Task scheduling using cron expression in @Scheduled annotation

Task scheduling using cron expression from properties file and @Scheduled annotation

Task scheduling using cron expression configured in context configuration

Sourcecode download link

**Explaining @Scheduled annotation**

This annotation is used for task scheduling. The trigger information needs to be provided along with this annotation. You can use the properties fixedDelay/fixedRate/cron to provide the triggering information.

1. **fixedRate** makes Spring run the task on periodic intervals even if the last invocation may be still running.
2. **fixedDelay** specifically controls the next execution time when the last execution finishes.
3. **cron** is a feature originating from Unix cron utility and has various options based on your requirements.

Example usage can be as below:

@Scheduled(fixedDelay =30000)

public void demoServiceMethod () {... }

@Scheduled(fixedRate=30000)

public void demoServiceMethod () {... }

@Scheduled(cron="0 0 \* \* \* \*")

public void demoServiceMethod () {... }

To use @Scheduled in your spring application, you must first define below xml namespace and schema location definition in your application-config.xml file.

xmlns:task="http://www.springframework.org/schema/task"

http://www.springframework.org/schema/task

http://www.springframework.org/schema/task/spring-task-3.0.xsd

Above additions are necessary because we will be using annotation based configurations. Now add below definition to enable annotations.

<task:annotation-driven>

Next step is to create a class and a method inside the class like below:

public class DemoService

{

@Scheduled(cron="\*/5 \* \* \* \* ?")

public void demoServiceMethod()

{

System.out.println("Method executed at every 5 seconds. Current time is :: "+ new Date());

}

}

Using @Scheduled annotation would in turn make Spring container understand that the method underneath this annotation would run as a job.  Remember that the methods annotated with @Scheduled should not have parameters passed to them. They should not return any values too. If you want the external objects to be used within your @Scheduled methods, you should inject them into the DemoService class using autowiring rather than passing them as parameters to the @Scheduled methods.

**Method 1) Task scheduling using fixed delay attribute in @Scheduled annotation**

In this method, ***fixedDelay*** attribute is used with @Scheduled annotation. Alternatively, ***fixedRate***can also be used.

A sample class will look like this:

package com.howtodoinjava.service;

import java.util.Date;

import org.springframework.scheduling.annotation.Scheduled;

public class DemoServiceBasicUsageFixedDelay

{

@Scheduled(fixedDelay = 5000)

//@Scheduled(fixedRate = 5000)

public void demoServiceMethod()

{

System.out.println("Method executed at every 5 seconds. Current time is :: "+ new Date());

}

}

And application configuration will look like this:

< ?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:task="http://www.springframework.org/schema/task"

xmlns:util="http://www.springframework.org/schema/util"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd

http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.0.xsd">

<task:annotation-driven />

<bean id="demoServiceBasicUsageFixedDelay" class="com.howtodoinjava.service.DemoServiceBasicUsageFixedDelay"></bean>

</beans>

**Method 2) Task scheduling using cron expression in @Scheduled annotation**

In this method, cron attribute is used with @Scheduled annotation. Value of this attribute must be a cron expression.

A sample class will look like this:

package com.howtodoinjava.service;

import java.util.Date;

import org.springframework.scheduling.annotation.Scheduled;

public class DemoServiceBasicUsageCron

{

@Scheduled(cron="\*/5 \* \* \* \* ?")

public void demoServiceMethod()

{

System.out.println("Method executed at every 5 seconds. Current time is :: "+ new Date());

}

}

And application configuration will look like this:

< ?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:task="http://www.springframework.org/schema/task"

xmlns:util="http://www.springframework.org/schema/util"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd

http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.0.xsd">

<task:annotation-driven />

<bean id="demoServiceBasicUsageCron" class="com.howtodoinjava.service.DemoServiceBasicUsageCron"></bean>

</beans>

**Method 3) Task scheduling using cron expression from properties file**

In this method, cron attribute is used with @Scheduled annotation. Value of this attribute must be a cron expression as in previous method, BUT, this cron expression will be defined in a properties file and key of related property will be used in @Scheduled annotation.

This will decouple the cron expression from source code, thus making changes easy.

A sample class will look like this:

package com.howtodoinjava.service;

import java.util.Date;

import org.springframework.scheduling.annotation.Scheduled;

public class DemoServicePropertiesExample {

@Scheduled(cron = "${cron.expression}")

public void demoServiceMethod()

{

System.out.println("Method executed at every 5 seconds. Current time is :: "+ new Date());

}

}

And application configuration will look like this:

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:task="http://www.springframework.org/schema/task"

xmlns:util="http://www.springframework.org/schema/util"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd

http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.0.xsd">

<task:annotation-driven />

<util:properties id="applicationProps" location="application.properties" />

<context:property-placeholder properties-ref="applicationProps" />

<bean id="demoServicePropertiesExample" class="com.howtodoinjava.service.DemoServicePropertiesExample"></bean>

</beans>

**Method 4) Task scheduling using cron expression configured in context configuration**

In this method, cron expression is configured in properties file, and job scheduling is configured in configuration file using property key for cron expression. Major change is that you do not need to use @Scheduled annotation on any method. Method configuration is also done in application configuration file.

A sample class will look like this:

package com.howtodoinjava.service;

import java.util.Date;

public class DemoServiceXmlConfig

{

public void demoServiceMethod()

{

System.out.println("Method executed at every 5 seconds. Current time is :: "+ new Date());

}

}

And application configuration will look like this:

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:task="http://www.springframework.org/schema/task"

xmlns:util="http://www.springframework.org/schema/util"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd

http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.0.xsd">

<task:annotation-driven />

<util:properties id="applicationProps" location="application.properties" />

<context:property-placeholder properties-ref="applicationProps" />

<bean id="demoServiceXmlConfig" class="com.howtodoinjava.service.DemoServiceXmlConfig" />

<task:scheduled-tasks>

<task:scheduled ref="demoServiceXmlConfig" method="demoServiceMethod" cron="#{applicationProps['cron.expression']}"></task:scheduled>

</task:scheduled-tasks>

</beans>