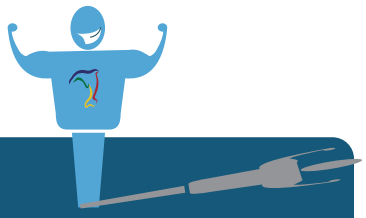
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

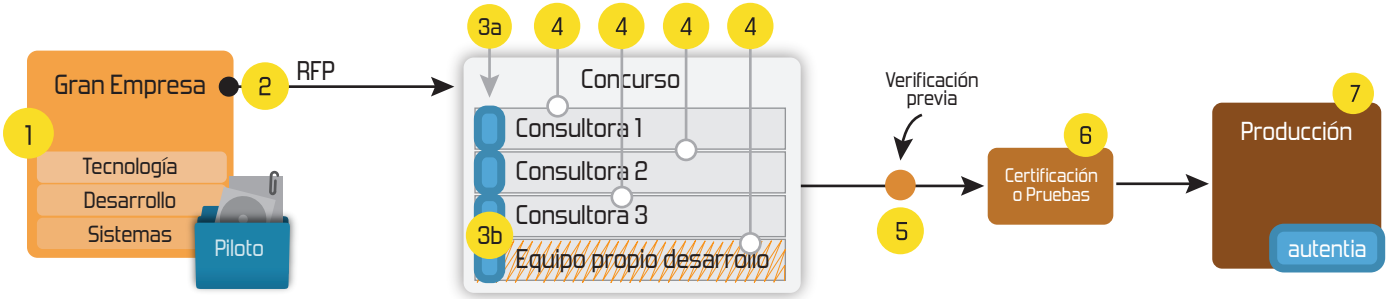
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Control de autenticación y acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com



autentia
real business solutions
Hosting patrocinado por
enredados



DESARROLLADO POR:
Jose Manuel Sánchez Suárez

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE



Fecha de publicación del tutorial: 2011-08-01

Share | Regístrate para votar

Spring schedule annotations.

0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno.
- 3. Dependencias.
- 4. Configuración vía applicationContext.
- 5. Haciendo uso de anotaciones.
- 6. Referencias.
- 7. Conclusiones.

1. Introducción

Tampoco ha llovido tanto, o si, desde que Juan, nos enseñara como configurar tareas programadas con el soporte de spring y quartz. El caso es que a partir de la versión 3 de spring la configuración se ha reducido sensiblemente y además se puede configurar vía anotaciones.

En este tutorial, vamos a ver cómo llevar a cabo dicha configuración de una manera más simple, haciendo uso de la misma clase de ejemplo del tutorial de referencia de Juan.

2. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 17' (2.93 GHz Intel Core 2 Duo, 4GB DDR3 SDRAM).
- Sistema Operativo: Mac OS X Snow Leopard 10.6.1
- Spring 3.1 M1
- Quartz 1.5.2

3. Dependencias.

Para hacer uso del soporte de tareas programadas de spring necesitaremos incluir en nuestro proyecto las siguientes dependencias, como siempre, lo mostramos haciendo uso de maven:

```
<dependencies>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>3.1.0.M1</version>
    </dependency>
    <dependency>
        <groupId>quartz</groupId>
        <artifactId>quartz</artifactId>
        <version>1.5.2</version>
    </dependency>
```

Para nuestras pruebas en el entorno de tests con jUnit y el soporte de inyección de spring, serán necesarias las siguientes:

Catálogo de servicios Autentia

Get ADOBE FLASH PLAYER

Últimas Noticias

- Pirineos on Tour
- VII Autentia Cycling Day
- Autentia patrocina la charla sobre Java SE 7 en Madrid
- Alfresco Day 2011
- XVII Charla Autentia - Grails - Vídeos y Material

Histórico de NOTICIAS

Últimos Tutoriales

- Primeros pasos con Google+
- Spring Security, Single Sign-On bajo plataformas Windows con Active Directory y Kerberos V5
- Spring MVC: acceder a las propiedades de un fichero desde una JSP con Expression Language (EL)
- Framework Scala liftweb
- Trabajando con JAXB y Eclipse

Últimos Tutoriales del Autor

- Expression Language 2.2 en Tomcat 6: invocar a un método de un ManagedBean en JSF pasando parámetros.
- Spring cache abstraction.
- Búsquedas "facetadas" en Solr con el soporte de Solrj.
- Indexación y recuperación de documentos en Apache Solr haciendo uso del api para Java.
- Introducción a Apache Solr.

Síguenos a través de:

Web2PDF
converted by Web2PDFConvert.com

```
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.8.2</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-test</artifactId>
        <version>3.1.0.M1</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

4. Configuración vía applicationContext.

Si llevamos a cabo la configuración a través del fichero de configuración de spring, sin anotaciones, sería la siguiente:

```
<?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:context="http://www.springframework.org/schema/context"
xmlns:p="http://www.springframework.org/schema/p"
    xmlns:task="http://www.springframework.org/schema/task"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task.xsd">


    <!-- Configuración de tareas programadas -->
    <bean id="memoryViewer" class="com.autentia.tutoriales.springschedule.MemoryViewer"/>

    <task:scheduled-tasks>
        <task:scheduled ref="memoryViewer" method="checkMemory" fixed-delay="500"/>
    </task:scheduled-tasks>

</beans>
```

Si la comparamos con las de versiones anteriores veis que se reduce sensiblemente. Equivaldría a lo que antes configurábamos de la siguiente forma:

```
<!-- Configuración de tareas programadas -->
<bean id="memoryViewer" class="com.autentia.tutoriales.springschedule.MemoryViewer"/>

<!-- Clase de Spring que se encarga de la llamada al metodo checkMemory de la clase MemoryViewer -->
<bean id="jobDetail" class="org.springframework.scheduling.quartz.MethodInvokingJobDetailFactoryBean">
  <property name="targetObject" ref="memoryViewer" />
  <property name="targetMethod" value="checkMemory" />
</bean>

<!-- Configuracion de la planificacion de la tarea -->
<bean id="simpleTrigger" class="org.springframework.scheduling.quartz.SimpleTriggerBean">
    <property name="jobDetail" ref="jobDetail" />
    <!-- Empieza pasados 10 segundos -->
    <property name="startDelay" value="10000" />
    <!-- Repite cada 5 segundos -->
    <property name="repeatInterval" value="5000" />
</bean>

<!-- Factoria de creacion del bean encargado de realizar la planificacion de la tarea -->
<bean class="org.springframework.scheduling.quartz.SchedulerFactoryBean">
    <property name="triggers">
        <ref bean="simpleTrigger" />
    </property>
</bean>
```

Ahora disponemos de un espacio de nombres propio para las tareas programadas que nos permiten realizar el siguiente tipo de configuraciones:

```
<task:scheduled-tasks>
    <task:scheduled ref="dailyRepairRepository" method="clearLastWeekRepairCache" cron="0 0 2 ? * MON"/>
    <task:scheduled ref="dailyRepairRepository" method="clearLastMonthRepairCache" cron="0 30 2 1 * ?"/>
    <task:scheduled ref="volumeRepairReportProcessor" method="processNoProcessedReports" fixed-delay="30000"/>
    <task:scheduled ref="activityReportProcessor" method="processNoProcessedReports" fixed-delay="30000"/>
</task:scheduled-tasks>
```

Donde el atributo ref no es más que un bean de spring, el atributo method apunta a un método del bean antes dicho y podemos programar la periodicidad con una cronExpression o un delay.

5. Haciendo uso de anotaciones.

Depende de nuestras preferencias, podemos hacer uso de anotaciones para configurar directamente en las propias clases de sevicio la ejecución de métodos como tareas programadas, incluyendo la cronExpression o el delay en la propia anotación.

Para activar la configuración vía anotaciones debemos prescindir de lo anterior e incluir simplemente la siguiente configuración:

```
<?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:context="http://www.springframework.org/schema/context"
xmlns:p="http://www.springframework.org/schema/p"
    xmlns:task="http://www.springframework.org/schema/task"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task.xsd">

    <context:annotation-config />
```


Últimas ofertas de empleo

2011-07-06
Otras Sin catalogar - LUGO.

2011-06-20
Comercial - Ventas - SEVILLA.

2011-05-24
Contabilidad - Expecialista Contable - BARCELONA.

2011-05-14
Comercial - Ventas - TARRAGONA.

2011-04-13
Comercial - Ventas - VALENCIA.


Jose Manuel Sánchez
sanchezsuarezj

Profe, no va el envío de imágenes en los emails!, incluyo el código pero aparece en verde y no se ejecuta. Anda, prueba a descomentarlo!! ;)
19 days ago · reply · retweet · favorite

Expression Language 2.2 en Tomcat 6: invocar a un método de un ManagedBean en #JSF pasando parámetros: http://bit.ly/qX2K8B
21 days ago · reply · retweet · favorite

#Autentia de boda. Enhorabuena Juan y Noelia !!!
23 days ago · reply · retweet · favorite

Comiendo en la panza es primero ;)

twitter
Join the conversation

Web2PDF
converted by Web2PDFConvert.com

```
    <context:component-scan base-package="com.autentia.tutoriales" />

    <task:annotation-driven/>
```

Con ello, podemos añadir las siguientes anotaciones directamente en nuestras clases de servicio:

```
@Service
public class MemoryViewer {

    @Scheduled(fixedDelay=500)
    public void checkMemory() {
        ...
    }

    @Scheduled(cron="0 0 2 ? * MON")
    public void _checkMemory() {
        ...
    }
}
```

6. Referencias.

- http://blog.springsource.com/2010/01/05/task-scheduling-simplifications-in-spring-3-0/

7. Conclusiones.

Reducimos la configuración, lo que redunda en claridad y mantenibilidad.

Un saludo.

Jose

jmsanchez@autentia.com

Anímate y coméntanos lo que pienses sobre este **TUTORIAL:**

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» **Registrate** y accede a esta y otras ventajas «

COMENTARIOS

SOME RIGHTS RESERVED Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

Copyright 2003-2011 © All Rights Reserved | Texto legal y condiciones de uso | Banners | Powered by Autentia | Contacto

XHTML 1.0 CSS XML RSS XML ATOM

Web2PDF
converted by Web2PDFConvert.com