

¿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

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

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

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



Hosting patrocinado por

[Inicio](#)[Quiénes somos](#)[Tutoriales](#)[Formación](#)[Empleo](#)[Colabora](#)[Comunidad](#)[Libro de Visitas](#)[Comic](#)

NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha aquí...

[Ver cursos que ofrece Autentia](#)[Descargar comics en PDF y alta resolución](#)[¡NUEVO!]
2008-02-19

2008-02-17 2008-02-14 2008-02-13

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la sección comic.

Tutorial desarrollado por



Raúl Expósito Díaz

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática de Gestión por la Universidad de Alcalá e Ingeniero en Informática por la Universidad Carlos III de Madrid. Perfil XING

Puedes encontrarme en Autentia

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

Descargar (6,2 MB)

Descargar en versión comic (17 MB)

AdictosAlTrabajo.com es el Web de difusión de conocimiento de Autentia.



Catálogo de cursos

Descargar este documento en formato PDF: logspringaop.pdf

Fecha de creación del tutorial: 2008-02-22

Logging de una aplicación Java mediante Spring AOP

1. Introducción

Al desarrollar aplicaciones debemos tener en mente que los mensajes de log van a ser imprescindibles una vez nuestra aplicación se encuentre en producción, ya que si ésta fallase necesitaremos una traza completa de mensajes de log que nos indiquen qué estaba haciendo la aplicación en el momento de fallar. Sin esos mensajes y sin las trazas de error puede ser muy difícil o incluso imposible averiguar qué ha pasado realmente para que nuestra aplicación haya fallado.

No sólo eso, sino que durante el desarrollo también deberemos controlar la ejecución de la aplicación mientras ésta se construye, asegurándonos de que funciona según lo esperado. Es por ello que existen varias técnicas con las que realizar tareas de depuración de código una vez éste se ha ejecutado, entre ellas las siguientes:

1. Usar puntos de ruptura y ejecutar paso a paso.
2. Imprimir mensajes de depuración en la consola.
3. Usar proxys dinámicos
4. Usar herramientas de profiling.
5. Emplear aspectos.

Sin embargo, ¿son estas técnicas excluyentes entre sí? No necesariamente, podríamos decir que son complementarias, ya que durante el desarrollo puede ser interesante poner, en momentos puntuales, un punto de ruptura o ejecutar paso a paso mientras que en producción esto es impensable.

En este tutorial nos centraremos en la 5ª de las opciones, utilizando para ello Spring AOP. En adictosaltrabajo hemos hablado otras veces de Spring y de aspectos (AOP). En el tutorial voy a usar como ejemplo un DAO, pero podríamos estar logueando cualquier tipo de clase.

¿Qué ventajas vamos a tener al seguir las trazas con AOP? Varias, entre las que cabe destacar que **no tenemos que poner en el código ninguna traza**, con lo que al programar podremos centrarnos únicamente en lo que debe hacer el código, "olvidándonos" de qué trazas debe mostrar.

La lista de técnicas la he obtenido de aquí, llegando hasta ahí desde esta noticia de javahispano.

Catálogo de servicios Autentia (PDF 6,2MB)



En formato comic...



☐ Web
☒ www.adictosaltrabajo.com

Buscar

Últimos tutoriales

2008-02-22
Crear un logger utilizado a través de aspectos con Spring AOP.

2008-02-20
Primeros pasos con PostgreSQL en Debian

2008-02-17
Cómo realizar pruebas unitarias con Spring y JUnit4 utilizando Gienah

2008-02-15
Creación de una aplicación con Spring e Hibernate desde 0

2008-02-07
Slimming básico de JBoss

2008-02-13
HtmlEmail. Envío de emails en HTML con imágenes embebidas

2008-02-11
Ireport con ODBC

2008-02-11
JUnit (3.8 y 4) y como ejecutar en un orden determinado los métodos de test de una clase de test

2008-02-10
Log4j: Cómo crear distintos logs en función de su naturaleza y nivel

2008-02-08
Tutorial de la API de Google Maps

2. Entorno

- Debian GNU/Linux 4.1 (Lenny)
- JDK 6 Update 1
- Spring 2.5

3. ¿Qué necesitamos?

Como vereis necesitamos poca cosa:

1. Un bean normal y corriente de Spring (en este ejemplo un DAO)
2. Un interceptor
3. Hacer un par de cambios en el fichero de contexto de Spring

La idea está en que al invocar al DAO, en realidad invoquemos al interceptor. El interceptor ejecutará el código que queramos, ejecutará el método del DAO que se desee ejecutar, y tras eso ejecutará más código que queramos (donde obviamente entre "el código que queramos" podremos poner una llamada al logger ;-))

3.1 El bean normal y corriente de Spring

Este bean será la "víctima" del interceptor. Como se puede ver es un bean normal y corriente.

```
view plain print ?
01. <bean id="autentiaDao" class="com.autentia.logspringaop.dao.impl.7
    <property name="sessionFactory" ref="sessionFactory" /></bean>
03.
```

3.2 El interceptor

Esta será la clase que hará cosas antes y después de ejecutar los métodos del objeto interceptado.

```
view plain print ?
01. import org.aopalliance.intercept.MethodInterceptor;import org.aop
02. org.apache.commons.logging.LogFactory;
03. /**
04.  * Clase encargada de interceptar las llamadas al DAO para logear
05.  */
06. public class DAOLoggerInterceptor implements MethodInterceptor {
07.
08.     private final static Log logger = LogFactory.getLog(DAOLoggerIr
09. );
10.
11.     public Object invoke(final MethodInvocation methodInvocation) th
12.     Throwable {
13.
14.         final long startTime = System.currentTimeMillis();
15.         try
16.         {
17.             return methodInvocation.proceed();
18.         }
19.         finally
20.         {
21.
22.             // generamos la lista de argumentos que recibe el metodo separados
23.             String arguments = new String();
24.             for (int i = 0; i < methodInvocation.getArguments().length; i++)
25.                 arguments += methodInvocation.getArguments()[i] + "
26.             }
27.
28.             // el metodo recibe al menos un argumento quitamos el e
29.             if (arguments.length() > 0) {
30.                 arguments = arguments.substring(0, arguments.length()
31.             }
32.
33.             logger.debug("[ " + (System.currentTimeMillis() - startT
34.                 methodInvocation.getMethod().getDeclaringClass()
35.                 methodInvocation.getMethod().getName() + " " +
36.                 arguments + " )");
37.         }
38.     }
39. }
```

En este código de ejemplo, el interceptor:

1. Antes de invocar al método, obtiene el número de milisegundos desde 1900 para poder mostrar en la traza cuánto tiempo ha tardado el método en ejecutarse.
2. Ejecuta el método objetivo
3. Muestra el número de milisegundos que ha tardado el método en ejecutarse seguido del nombre de la clase, del nombre del método y de los parámetros recibidos por el método.

Gracias a este interceptor podemos quitar el logger del DAO porque ya no hace falta. Cuando se invoque cualquiera de sus métodos se lanzará el interceptor que ya se encargará del log en lugar de hacerlo el DAO. Con ello el código del DAO queda más limpio.

En este tutorial sólo hay un DAO y un interceptor, pero podría haber más DAO y clases en general que utilizasen el mismo interceptor. Fijaros en que el interceptor extiende de la clase `MethodInterceptor`

últimos cambios de empleo

2008-02-06
T. Información -
Analista / Programador
- MADRID.

2008-02-04
T. Información -
Becario - MADRID.

2008-01-28
T. Información -
Becario - MADRID.

2008-01-25
Otras Sin catalogar -
MURCIA.

2008-01-24
T. Información -
Analista / Programador
- MADRID.

3.3 Cambios en el contexto de Spring

Lo único que falta ahora es configurar el contexto para que se ejecute el interceptor antes que la clase interceptada. Para ello deberemos hacer los siguientes cambios:

- Declarar el interceptor, que como podeis ver es muy sencillo:

```
view plain print ?
01. <bean id="daoLoggerInterceptor" class="com.autentia.logsprin
02.
```

- Remembrar el viejo DAO, poniéndole el sufijo 'Target':

```
view plain print ?
01. <bean id="autentiaDaoTarget" class="com.autentia.logspringao
    <property name="sessionFactory" ref="sessionFactory" /></be
03.
```

- Crear un nuevo bean que se llame igual que el interceptado (autentiaDao) para que la aplicación pase a usar el nuevo en vez del viejo:

```
view plain print ?
01. <bean id="autentiaDao"
    class="org.springframework.aop.framework.ProxyFactoryB
03.     <property name="proxyInterfaces">
        <value>com.autentia.logspringaop.dao.impl.Autent
05.     </property>
06.     <property name="interceptorNames">
07.         <list>
08.             <value>daoLoggerInterceptor</value>
09.             <value>autentiaDaoTarget</value>
10.         </list>
11.     </property>
12. </bean>
13.
```

Fijaros en los 3 campos <value>, y en que en ellos se pone la clase DAO objetivo, el interceptor, y el nuevo nombre que tenga el antiguo DAO.

Tras haber hecho estos cambios ya disfrutaremos de un DAO cuyas trazas de log se escriben mediante AOP.

4. Conclusiones

Como ya he ido comentando a lo largo del tutorial, el uso de aspectos para mostrar trazas de log me parece especialmente cómodo ya que al escribir código no hay que preocuparse del log, dejando el código más limpio y permitiendo que los log no distraigan de su lectura.

Aparte, tal y como hemos visto en el tutorial, podemos acceder a todo tipo de información del método, como por ejemplo qué parámetros recibe. También podríamos haber cubierto casos en los cuales los métodos lancen excepciones, pero lo dejaremos para otro tutorial :-)

También hemos podido ver que es muy sencillo de configurar, ya que sólo deberemos hacer un par de cambios en el contexto de Spring. En definitiva, es una opción muy recomendable.

Espero que os sea de utilidad.

- Puedes opinar sobre este tutorial haciendo clic aquí.
- Puedes firmar en nuestro libro de visitas haciendo clic aquí.
- Puedes asociarte al grupo AdictosAlTrabajo en XING haciendo clic aquí.
- Añadir a favoritos Technorati.



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

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido (Ver todos los tutoriales). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

info@autentia.com

Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de suscripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Comparativa entre EJB3 y Spring	En este tutorial os mostramos una comparativa entre EJB3 y Spring esperando que os ayude a decidir qué tecnología utilizar.	2007-10-17	1996	pdf
URLs amigables con Spring MVC	En este tutorial se va a hacer un ejemplo práctico utilizando Spring MVC para la configuración de URLs amigables de nuestra aplicación	2007-04-11	4219	pdf
SpringIDE, plugin de Spring para Eclipse	En adictosaltrabajo os hemos ido presentando diversos plugins para Eclipse. Esta vez le toca el turno a SpringIDE, un plugin que os ayudará a desarrollar aplicaciones que utilicen Spring.	2008-01-19	865	pdf
Cómo realizar pruebas unitarias con Spring y JUnit4 utilizando Gienah	En este tutorial vamos a presentaros Gienah, una tecnología que os permitirá de una forma muy cómoda y sencilla utilizar componentes de Spring en vuestros test unitarios realizados con JUnit 4	2008-02-17	145	pdf
Spring WebFlow Tiles	En este tutorial aprenderemos el uso de tiles para usarlo en conjunción con Spring WebFlow.	2007-11-26	921	pdf
Manual básico de Spring WebFlow	En este tutorial Javier Antoniucci nos enseña cómo empezar a trabajar con el framework de desarrollo web Spring webflow.	2007-11-26	1356	pdf
Spring WebFlow con Validator	En este tutorial se muestra como podemos realizar las validaciones más frecuentes de datos mediante Spring WebFlow.	2007-12-11	861	pdf
Spring: definición dinámica de Beans	Este tutorial habla sobre la modificación dinámica de los beans del contexto para simplificar la configuración de Spring	2007-05-09	2853	pdf
Creación de una aplicación con Spring e Hibernate desde 0	Este tutorial vamos a explicar paso a paso cómo crear una pequeña aplicación usando Spring e Hibernate con anotaciones partiendo desde 0	2008-02-15	462	pdf
Introducción a Spring Web Flow	Spring Web Flow es un módulo de extensión del framework Spring, que facilita la implementación del flujo de páginas de una aplicación web	2006-01-03	14724	pdf

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.