

¿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)



Powered by **autentia** Hosting patrocinado por **enredados**

- [Inicio](#)
- [Quienes 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!] 2008-04-14



2008-04-07



2008-04-01



2008-03-25

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la seccion [comic](#).

Tutorial desarrollado por



Carlos García Pérez

Creador del pionero Web **MobileTest**.

Consultor tecnológico en el desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática *

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: [smtpLog4j.pdf](#)

Fecha de creación del tutorial: 2008-04-19

Log4J, SMTPAppender: Envío de trazas de log por email

Una tarea fundamental en todo proyecto informático es la generación de log o trazas, generalmente para detectar y solucionar los posibles errores que pudieran producirse.

Aunque normalmente estos logs suelen dejarse en archivos de texto, en determinadas situaciones puede ser necesario enterarse rápidamente de los errores sin tener que esperar a que el usuario nos lo notifique. Una posible solución podría ser que el sistema nos enviase esos errores a una cuenta de correo electrónico.

En este tutorial vamos a aprender como hacerlo, para ello nos basaremos en una sencilla aplicación que genera errores por correo electrónico cuando una simulada compra de productos falla.

BuyCazd: Simula un carrito de compra.

Catálogo de servicios Autentia (PDF 6,2MB)



[En formato comic...](#)



Web
 www.adictosaltrabajo.com

Últimos tutoriales

2008-04-19
[Log4J, SMTPAppender: Envío de trazas de log por email](#)

2008-04-17
[Indentación del código fuente](#)

2008-04-16
[Explorar bases de datos HSQldb](#)

2008-04-09
[Crear formularios con Google Docs](#)

2008-04-09
[Trident IDE, una herramienta para generar contenidos SCORM](#)

2008-04-08
[E-learning y SCORM @](#)

2008-04-08
[Comparativa de antivirus freeware y opensource](#)

2008-04-05
[JMX y monitorización de JBoss](#)

2008-04-05
[Jersey: la implemetación de RESTFull de Sun](#)

2008-04-05
[Metro: pila de webservices de Sun. Integración con Maven 2](#)

Últimas ofertas de empleo

2008-04-14
[T. Información - Analista / Programador - MADRID.](#)

```

view plain print ?
01. package com.autentia.tutoriales.log4j;
02.
03.
04. import java.util.ArrayList;
05. import java.util.Random;
06.
07. import org.apache.commons.logging.Log;
08. import org.apache.commons.logging.LogFactory;
09.
10.
11. /**
12.  * Carrito de compra
13.  * @author Carlos García. Autentia
14.  * @see http://www.mobiletest.es
15.  */
16. public class BuyCard {
17.     private static Log log = LogFactory.getLog(BuyCard.class);
18.     private java.util.ArrayList<String> pedido;
19.
20.
21.     /**
22.      * Constructor
23.      */
24.     public BuyCard(){
25.         this.pedido = new ArrayList<String>();
26.     }
27.
28.     /**
29.      * Añade un producto al carrito de la compra
30.      * @param code Código del producto
31.      * @throws java.util.NoSuchElementException En caso de que el producto no exista o r
32.      */
33.     public void appendProduct(String code) throws java.util.NoSuchElementException {
34.         if (!this.isAvailable(code)){
35.             if (log.isDebugEnabled()){
36.                 log.debug("Se ha solicitado un producto que no existe: " + code);
37.             }
38.             throw new java.util.NoSuchElementException(code);
39.         }
40.
41.         pedido.add(code);
42.
43.         if (log.isDebugEnabled()){
44.             log.debug("El producto ha sido agregado al carrito correctamente: " + code);
45.         }
46.     }
47.
48.     /**
49.      * @param code Código del producto
50.      * @return Indica un producto está en stock
51.      */
52.     public boolean isAvailable(String code){
53.         Random rnd = new Random();
54.         int nRnd = rnd.nextInt(100); // Generamos número aleatorio entre 0-100
55.         boolean exists = (nRnd >= 5); // Probabilidad baja de que el producto no exista.
56.
57.         if (log.isDebugEnabled()){
58.             log.debug("Producto " + code + ", dispobibilidad: " + exists);
59.         }
60.
61.         return exists;
62.     }
63.
64.     /**
65.      * Realiza el proceso de compra
66.      */
67.     public boolean processBuy() {
68.         if (log.isDebugEnabled()){
69.             log.debug("Realizando la compra");
70.         }
71.
72.         Random rnd = new Random();
73.         int nRnd = rnd.nextInt(100); // Generamos número aleatorio entre 0-100
74.         boolean ok = (nRnd >= 50); // 50% de probabilidad de que el proceso de compra cc
75.
76.         if (log.isDebugEnabled()){
77.             log.debug("compra realizada correctamente: " + ok);
78.         }
79.
80.         return ok;
81.     }
82. }

```

2008-04-10
T. Información - Analista /
Programador - BARCELONA.

2008-04-04
Banca - Genérico - MADRID.

2008-04-03
Banca - Genérico - MADRID.

2008-04-02
T. Información - Analista /
Programador - MADRID.

Anuncios Google

La siguiente aplicación de consola simula un proceso de compra de varios productos.

```

view plain print ?
01. package com.autentia.tutoriales.log4j;
02.
03. import org.apache.commons.logging.Log;
04. import org.apache.commons.logging.LogFactory;
05. import org.apache.log4j.xml.DOMConfigurator;
06.
07. /**
08.  * Ejemplo del tutorial
09.  * @author Carlos García. Autentia.
10.  * @see http://www.mobiletest.es
11.  */
12. public class EjemploLog {
13.
14.     /**
15.      * Punto de entrada a la aplicación
16.      */
17.     public static void main(String[] args) {
18.         // Cargamos la configuración de Log4J desde un archivo accesible via CLASSPATH
19.         DOMConfigurator.configure(EjemploLog.class.getClassLoader().getResource("autentiaLog4jSmtp.xml")
20.
21.         // Obtenemos el Log para esta clase.
22.         Log log = LogFactory.getLog(EjemploLog.class);
23.
24.         BuyCard compra = new BuyCard();
25.
26.         try {
27.             // Solicitamos unos productos
28.             compra.appendProduct("98HG");
29.             compra.appendProduct("73DD");
30.             compra.appendProduct("671A");
31.
32.             if (compra.processBuy()){
33.                 log.debug("La compra se ha realizado correctamente" );
34.             } else {
35.                 Log smtp = LogFactory.getLog("soporteCompras");
36.                 smtp.error("Se ha producido un error grave al realizar el proceso de compra" );
37.             }
38.         } catch (Exception ex){
39.             log.warn("Error en la compra:", ex);
40.         }
41.     }
42. }

```

Si observa, en las líneas 35 y 36 se ve como la aplicación selecciona el Logger `soporteCompras` para notificar el error por correo.

Dependencias necesarias para ejecutar la aplicación:

```

01. // Para el sistema de Logs necesitamos:
02. log4j-X.Y.jar
03. commons-logging-X.Y.jar
04.
05. // Para el envio de correos necesitamos:
06. mail-X.Y.jar
07. activation-X.Y.jar

```

Archivo de configuración de Log4j: `autentiaLog4jSmtp.xml`

```

view plain print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <!DOCTYPE log4j:configuration SYSTEM "dtds/log4j.dtd" >
03. <log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/" debug="false">
04.     <appender name="file" class="org.apache.log4j.RollingFileAppender" >
05.         <param name="File" value="autentiaLog4jApp.log" />
06.         <param name="MaxFileSize" value="2000000" />
07.         <param name="MaxBackupIndex" value="5" />
08.         <param name="Threshold" value="DEBUG" />
09.
10.         <layout class="org.apache.log4j.PatternLayout" >
11.             <param name="ConversionPattern" value="%nd{yyyy-MM-dd HH:mm:ss} [%-5p] [%1] %n%m%n" />
12.         </layout>
13.     </appender>
14.
15.     <appender name="email" class="org.apache.log4j.net.SMTPAppender" >
16.         <param name="SMTPHost" value="smtp_servidor.com" />
17.         <param name="SMTPUsername" value="tu_usuario" />
18.         <param name="SMTPPassword" value="tu_password_usuario" />
19.         <param name="From" value="noreply@tu_servidor.com" />
20.         <param name="To" value="cuenta_soporte@tu_servidor.com" />
21.         <param name="Subject" value="Notificación de la aplicación" />
22.
23.         <param name="Threshold" value="DEBUG" />
24.
25.         <layout class="org.apache.log4j.PatternLayout" >
26.             <param name="ConversionPattern" value="%nd{yyyy-MM-dd HH:mm:ss} [%-5p] [%1] %n%m%n" />
27.         </layout>
28.     </appender>
29.
30. <root>
31.     <appender-ref ref="file" />
32. </root>
33.
34. <logger name="soporteCompras" >
35.     <appender-ref ref="email" />
36. </logger>
37. </log4j:configuration >

```

Si observa en las líneas 16-21 debemos poner la configuración necesaria para realizar la conexión al servidor SMTP y enviar el correo electrónico.

En el archivo `autentiaLog4jApp.log` la aplicación deja todas las trazas que genera la aplicación.

```

01. 2008-04-18 16:18:01 [DEBUG] [com.autentia.tutoriales.log4j.BuyCard.isAvailable(BuyCard.java: 58)]
02. Producto 98HG, disponibilidad: true
03.
04. 2008-04-18 16:18:01 [DEBUG] [com.autentia.tutoriales.log4j.BuyCard.appendProduct(BuyCard.java: 44)]
05. El producto ha sido agregado al carrito correctamente: 98HG
06.
07. 2008-04-18 16:18:01 [DEBUG] [com.autentia.tutoriales.log4j.BuyCard.isAvailable(BuyCard.java: 58)]
08. Producto 73DD, disponibilidad: true
09.
10. 2008-04-18 16:18:01 [DEBUG] [com.autentia.tutoriales.log4j.BuyCard.appendProduct(BuyCard.java: 44)]
11. El producto ha sido agregado al carrito correctamente: 73DD
12.
13. 2008-04-18 16:18:01 [DEBUG] [com.autentia.tutoriales.log4j.BuyCard.isAvailable(BuyCard.java: 58)]
14. Producto 671A, disponibilidad: true
15.
16. 2008-04-18 16:18:01 [DEBUG] [com.autentia.tutoriales.log4j.BuyCard.appendProduct(BuyCard.java: 44)]
17. El producto ha sido agregado al carrito correctamente: 671A
18.
19. 2008-04-18 16:18:01 [DEBUG] [com.autentia.tutoriales.log4j.BuyCard.processBuy(BuyCard.java: 69)]
20. Realizando la compra
21.
22. 2008-04-18 16:18:01 [DEBUG] [com.autentia.tutoriales.log4j.BuyCard.processBuy(BuyCard.java: 77)]
23. compra realizada correctamente: false
24.
25. 2008-04-18 16:18:01 [ERROR] [com.autentia.tutoriales.log4j.EjemploLog.main(EjemploLog.java: 36)]
26. Se ha producido un error grave al realizar el proceso de compra

```

Además de la salida que se observa en el log, la aplicación habrá enviado un correo electrónico con la información que aparecen en las líneas 25 y 26.

Conclusiones

La gestión de trazas es una parte fundamental de toda aplicación, por lo que es recomendable establecer alguna política o pautas al respecto dentro del proyecto.

Recordar que la mayor parte del dinero de un sistema informático se va en mantenimientos, así que un buen sistema de logs nos hará la vida más fácil y nos permitirá ahorrar muchos esfuerzos (dinerito).

Referencias y enlaces de interés

- [Sitio Oficial del proyecto Log4J](#)
- [Log4J JavaDocs](#)
- [Tutorial: Introducción a Log4J](#)
- [Tutorial: Cómo crear distintos logs en función de su naturaleza y nivel](#)

Saludos,
Carlos García, creador de [MobileTest](#)

- 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 ...

Autentia = Soporte a Desarrollo & Formación.

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
Escritura log con Fichero UDP y JMS	Os mostramos ejemplos para cuantificar el coste de escritura de Logs por pantalla, fichero, UDP y JMS (describiendo como configurar el entorno)	2003-08-23	25376	pdf
Log4J: Cómo crear distintos logs en función de su naturaleza y nivel	Log4J: Cómo crear distintos logs en función de su naturaleza y nivel	2008-02-10	1349	pdf
Primeros pasos con Log 4 NET	En este tutorial se muestra como almacenar las trazas de una aplicación con ayuda de Log 4 Net.	2005-12-15	7638	pdf
Test con JUnit	Cuando se hacen desarrollo profesionales, no basta con hacer los programas, hay que asegurarse de que van a funcionar. Una de las técnicas más seguras es crear aplicaciones que incluyan el código para autoprobarse. Os mostramos como usar JUnit	2003-06-21	17276	pdf
Crear un logger utilizado a través de aspectos con Spring AOP.	En este tutorial os enseñamos cómo implementar un logger utilizado a través de aspectos con Spring AOP.	2008-02-22	892	pdf
Introducción a log4j	En un desarrollo Java es vital normalizar los logs para posteriormente poder depurar el funcionamiento de nuestra aplicación. Os mostramos como usar Log4J.	2003-06-13	29400	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.