

¿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)



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)

| | |
|---|--|
| <p>Tutorial desarrollado por: Francisco Javier Martínez Páez</p> <p>Puedes encontrarme en Autentia Somos expertos en Java/J2EE Contacta en info@autentia.com</p> | <p>www.adictosaltrabajo.com es el Web de difusión de conocimiento de www.autentia.com</p>  <p>autentia real business solutions</p> <p>Catálogo de cursos</p> |
|---|--|

Descargar este documento en formato PDF [NotificacionesMBean.pdf](#)

[Firma en nuestro libro de Visitas](#)

Master Java Certificado

Nuevo Temario-UML-JSF-AJAX-J2ME Trabajo
 .Garantizado-Bolsa de Empleo
www.exes.es

Cursos de informática

TECSUR Cursos de
 .Java, Linux, Oracle .Net, C, Autocad, Ofimática,
 915590611
www.tecsur.es

SOFTENG

Desarrollo soluciones web y gestión
 .Consultoría informática Barcelona.
www.softeng.es

[Anuncios Google](#)

[Anunciarse en este sitio](#)

Recibiendo las notificaciones de un XMBean

Los ejemplos de este tutorial están hechos con el siguiente entorno de desarrollo:

- Jboss Eclipse IDE Milestone 5.
- JDK 1.5
- JBoss 4.0.5 GA

INTRODUCCION.

Este tutorial es una continuación de dos anteriores llamados *MBeans* y *JBoss*:

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=MBeansJBoss>

y *XMBears* y *JBoss*:

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=XMBears>

En este último tutorial, os mostré como crear un XMBean y registrarlo en el servidor de MBeans de JBoss, así como añadirle la persistencia. Sin embargo, nos quedó pendiente hacer algún ejemplo de envío y recepción de notificaciones.

Ya os comenté que los MBeans envían notificaciones cuando se registran o cuando algún atributo es modificado, sin necesidad de añadir nada. Pero, nuestro objetivo es enviar notificaciones cuando alguien invoque el método `formatFecha` del XMBean que hicimos en el tutorial anterior. Vamos a modificar el código del método `formatFecha` XMBean *XFecha* para que envíe notificaciones:

```
public String formatFecha(String formatoActual) throws Exception {
    String formatoUsar = "";

    if(this.formato!=null && !"".equals(this.formato.trim())) {
        formatoUsar=this.formato.trim();
    } else {
        formatoUsar="dd/MM/yyyy";
    }

    if(formatoActual!=null && !"".equals(formatoActual.trim())) {
        formatoUsar = formatoActual.trim();
    }

    SimpleDateFormat formatter=new SimpleDateFormat(formatoUsar);
    String resultado = formatter.format(new Date());

    // Creamos un objeto javax.management.Notification
    Notification createMsg = new Notification("formatFecha",this,new Date().getTime());
    // Le añadimos a la notificacion informacion adicional.
    createMsg.setUserData("Formateada la fecha actual con el formato:" + formatoUsar + " con el resultado: " +
    resultado);

    // Enviamos la notificación.
    sendNotification(createMsg);

    return resultado;
}
```

Ahora, lo que necesitamos es alguien que escuche las notificaciones que se envían desde el nuestro Bean.

CREANDO EL MBEAN RECEPTOR

Para crear un MBean que reciba notificaciones (un **listener**), nuestro objetivo se crean un MBean que implemente el interfaz: *ListenerServiceMBean*.

Han de implementar este interfaz, todos aquellos MBeans que quieran recibir notificaciones. Alguno preguntará, ¿todas las notificaciones? No, solamente recibirá aquellas notificaciones a las que se suscriba (parecido a un topic JMS).

Además, para facilitarnos las cosas, al igual que ocurría con el interfaz *ServiceMBean* en el que Jboss nos regala una implementación del mismo en la clase *ServiceMBeanSupport*, ocurre que existe una implementación de *ListenerServiceMBean* en la clase: *ListenerServiceMBeanSupport*. Este consta de los siguientes métodos:

```
public void subscribe(boolean dynamicSubscriptions)
    throws Exception;
public void subscribe(boolean dynamicSubscriptions, ObjectName listener)
    throws Exception;
public void unsubscribe();

public void handleNotification2(Notification notification, Object handback);
```

Los tres primeros nos sirven para suscribirnos / darnos de baja de manera dinámica de alguna suscripción. El último de ellos es el que más nos interesa, ya que será el método de nuestro MBean que se invoque cuando se produzca alguna notificación a la que estemos suscritos (algo parecido al método *onMessage* de un MDB).

Dicho esto, lo primero que haremos es crearnos el interfaz: *XFechaListenerMBean*

```
package com.autentia.tutoriales.jboss.listener;
import org.jboss.system.ListenerServiceMBean;

public interface XFechaListenerMBean extends ListenerServiceMBean
{
}
```

Recordad que el nombre del interfaz debe acabar en MBean.

Ahora, nos crearemos la clase del Bean: *XFechaListener*:

```
package com.autentia.tutoriales.jboss.listener;

import javax.management.Notification;
import org.jboss.system.ListenerServiceMBeanSupport;

public class XFechaListener extends ListenerServiceMBeanSupport
    implements XFechaListenerMBean{

    public XFechaListener() {
    }

    /**
     * Este es el método que será invocado cuando se reciba una notificación de un MBean al que estemos suscritos. */
    public void handleNotification2(Notification notif, Object handback) {
        System.out.println("---- RECIBO UNA NOTIFICACION ----");
        System.out.println("notif.getType():"+notif.getType());
        System.out.println("notif.getTimestamp():"+notif.getTimestamp());
        System.out.println("notif..getUserData():"+notif.getUserData());

    }

    /**
     * Sobreescribimos el metodo startService para suscribirnos a las notificaciones cuando arranquemos */
    protected void startService() throws Exception {
        /* El valor true es para indicarle que si se registran nuevas instancias de MBeans en el servidor de MBeans que cumplen
        nuestros criterios de subscripcion, nos suscriban tambien a sus notificaciones (genial ¿no?) */
        System.out.println("STARTING LISTENER");
        super.subscribe(true);
    }

    /**
     * Para darnos de baja cuando se pare nuestro MBean.
     */
    protected void stopService() throws Exception {
        System.out.println("STOPPING LISTENER");
        super.unsubscribe();
    }

}
```

Ahora nos creamos el descriptor del listener (jboss-service.xml):

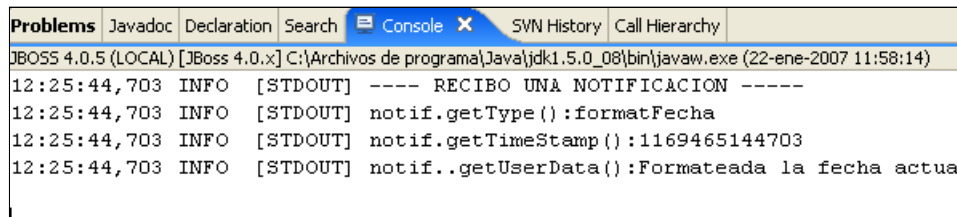
```
<?xml version="1.0" encoding="UTF-8"?>
<server>
<mbean code="com.autentia.tutoriales.jboss.listener.XFechaListener"
    name="autentia.tutoriales.paco:service=XFechaListener">
    <attribute name="SubscriptionList">
    <subscription-list>
    <mbean name="autentia.tutoriales.paco:*/>
    </subscription-list>
    </attribute>
</mbean>
</server>
```

En *SubscriptionList* le indicamos los MBeans de los que queremos recibir notificaciones. Nosotros le hemos indicado que queremos recibir cualquier notificación emitida por un MBean que esté bajo el dominio: *autentia.tutoriales.paco*

Una vez llegado aquí, creaos el ".sar", desplegado e id a la consola jmx:



Seleccionad el Bean XFormateadorFechaActual e invocad el método formatFecha y comprobad las trazas:



Comprobaréis que recibiréis la notificación de la invocación al método. También recibiréis notificaciones cuando se modifica el atributo formato o se inicia XFormateadorFechaActual.

Bueno, lo importante de todo esto es que se os ocurran ideas donde aplicar lo que os he enseñado en estos últimos tutoriales, ya que aprender a hacerlo es cuestión de mirar documentación.

Si nos necesitáis, ya sabéis donde buscarnos: <http://www.autentia.com>



This work is licensed under a [Creative Commons Attribution-Noncommercial-No Derivative Works 2.5 License](http://creativecommons.org/licenses/by-nc-nd/2.5/).



[Puedes opinar sobre este tutorial aquí](#)

Recuerda

que el personal de [Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

¿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?

info@autentia.com

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

Autentia = Soporte a Desarrollo & Formación

Creatividad Internet

[Autentia S.L.](#) Somos expertos en:

J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..
y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

| | |
|--------------------------|---------------------------------------|
| Subscribirse a Novedades | |
| e-mail | |
| | <input type="button" value="Enviar"/> |

Otros Tutoriales Recomendados ([También ver todos](#))

| Nombre Corto | Descripción |
|---|---|
| mod_jk en WindowsXP / Apache2-JBoss | Os mostramos como instalar el conector mod_jk sobre WindowsXP utilizando Apache2 y JBoss |
| Guía rápida de instalación de JBOSS Application Server 4. | En este manual veremos paso a paso la forma de instalar en tu equipo JBoss Application Server 4. |
| Comunicación entre TAGs, Beans y JSPs | Os mostramos las posibilidades de comunicación entre JSPs, Bean y etiquetas de usuario. |
| WebServices con Axis y JBoss | En este tutorial os mostramos como realizar servicios web utilizando Axis y el contenedor de aplicaciones web JBoss |
| mod_jk en Linux / Apache2-JBoss | El conector mod_jk se encarga de enviar las peticiones dinámicas de Apache2 a un servidor de aplicaciones JBoss |
| mod_jk en Ubuntu / Apache2-JBoss | Os mostramos como instalar el conector mod_jk sobre la distribución linux Ubuntu utilizando Apache2 y JBoss |
| mod_jk en WindowsXP / ISS-JBoss | Os mostramos como instalar el conector mod_jk sobre WindowsXP utilizando ISS y JBoss |
| Instalar JBoss | Os mostramos como instalar en servidor gratuito de aplicaciones JBOSS así como a automatizar su arranque y parada. |
| XMBeans y JBoss | Este tutorial es una continuación de una anterior llamado MBeans y JBoss |
| MBeans y JBoss | Este tutorial tiene como finalidad familiarizarse con el servidor de aplicaciones JBoss y su API JMX |

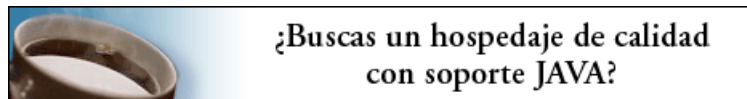
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)



www.AdictosAlTrabajo.com Optimizado 800X600