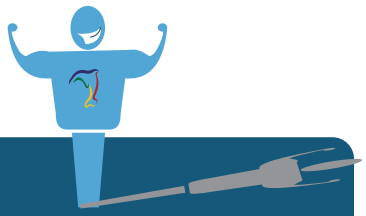
autentia

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...

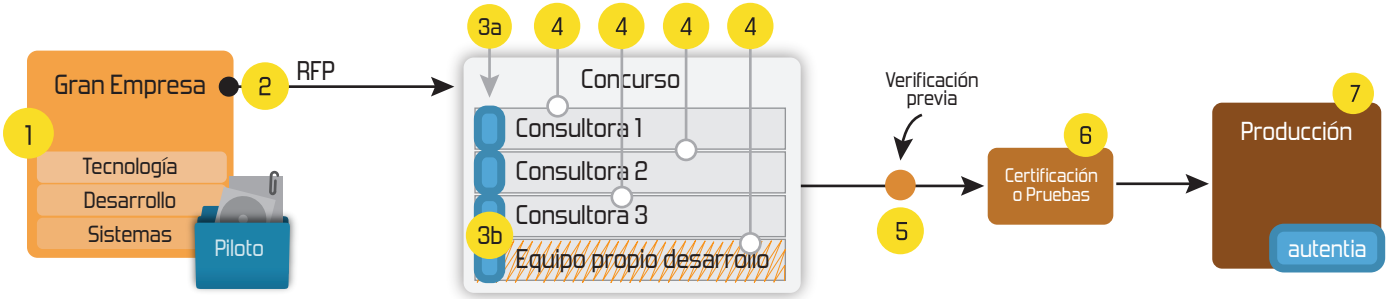
1. Desarrollo de componentes y proyectos a medida



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

Lanzado TNTConcept versión 0.8 (10/12/2007)

¿Gestionas tu empresa con hojas de cálculo? ¿No crees que puede haber un modo mejor?

Desde Autentia ponemos a vuestra disposición el software que hemos construido (100% gratuito, con código fuente disponible y sin restricciones funcionales) para nuestra gestión interna, llamado TNTConcept (au**T**e**NT**ia). Construida con las últimas tecnologías de desarrollo Java/J2EE (Spring, JSF, Acegi, Hibernate, Maven, Subversion, etc.) y disponible en licencia GPL, seguro que a muchos profesionales independientes y PYMES os ayudará a organizar mejor vuestra operativa.

Las cosas grandes empiezan siendo algo pequeño ..... Saber más en:
http://tntconcept.sourceforge.net/

Tutorial desarrollado por: Carlos García Pérez

Puedes encontrarme en Autentia
Somos expertos en Java/J2EE
Contacta en:
carlos.garcia@autentia.com

Creador de MobileTest

NUEVO CATÁLOGO DE SERVICIOS DE AUTENTIA (PDF 6,2MB)

www.adictosaltrabajo.com es el Web de difusión de conocimiento de www.autentia.com

autentia
real business solutions

Catálogo de cursos

Descargar este documento en formato PDF servicemonitor.pdf

Firma en nuestro libro de Visitas **<-----> Asociarme al grupo AdictosAlTrabajo en eConozco**

Virtual Software
Desarrollo de software Programas de gestion empresas
www.virtualsw.es

Master Experto Java
100% alumnos se colocan. Incluye Struts, Hibernate, Ajax
www.grupoatrium.com

PDF Libraries for Java
Generate, Merge, Append, Barcodes And So Much More, Free Evaluation!
www.dynamicPDF.com

KOTASOFT Software Factory
Desarrollo software CMMI nivel 3. Soluciones J2EE - Struts,Spring,...
www.kotasoft.com

Fecha de creación del tutorial: 2007-12-14

Monitorización de servicios: start, stop, restart, status

En este tutorial os voy a mostrar como lanzar un aplicación en modo servicio con las siguientes características:

- El servicio respondará a los mensajes: start, stop, restart, status.
- El servicio se iniciará automáticamente cuando se inicie la máquina.
- El servicio se iniciará automáticamente en caso de que sea detenido.

Como veis son muchas tareas las que vamos a realizar, asi que empezemos poco a poco.

Uno de los problemas que tenemos que resolver es saber cuando un servicio está obreciendo servicio a sus clientes. La mejor forma de saberlo es haciéndole una petición para ver si contexta.
Para este ejemplo, vamos a crear un Thread que será facilmente integrable con cualquier aplicación y que simplemente contexta una petición UDP sobre un puerto determinado retornando un "ok" al cliente, es decir algo parecido a un PING.

Código fuente que necesitamos integrar en nuestros servicios:

view plain print ?

```
01.
02. package com.autentia.tutoriales.servicemonitor;
03.
04. import java.net.*;
05.
06. /**
07.  * Esta clase recibe proporciona funcionalidad de monitorización del estado del servicio.
08.  * @author Carlos García. Autentia.
09.  */
10. public class StatusListener extends Thread {
11.
12.    private DatagramSocket socket;
13.
14.    /**
15.     * Constructor
16.     * @param listen_port Puerto en el que se desea escuchar
17.     * @throws SocketException En caso de que el socket no se pueda enlazar al puerto especificado
18.     */
19.    public StatusListener(int listen_port) throws SocketException {
20.       this.socket = new DatagramSocket(listen_port);
21.    }
22.
23.    /*
24.     * Iniciamos el proceso de monitorización de estado.
25.     * @see java.lang.Thread#run()
26.     */
27.    public void run(){
28.       byte[]        inbytes  = new byte[128];
29.       DatagramPacket request  = new DatagramPacket(inbytes, inbytes.length);
30.       DatagramPacket response = null;
31.       byte[]        ok       = "ok".getBytes();
32.
33.
34.       try {
35.          while (true){
36.             try {
37.                // Esperamos las peticiones de consulta de estado. (Es un método bloqueante)
38.                socket.receive(request);
39.
40.                // Hemos recibido una petición. Creamos la respuestsa
41.                response = new DatagramPacket(ok, 0, ok.length, request.getAddress(), request.getPort());
42.
43.
44.                // Enviamos la respuesta
45.                socket.send(response);
46.
47.             } catch (java.io.IOException ex){
48.                // Se ha producido un error en el tratamiento de la petición actual.
49.                // No pasa nada, seguimos dando servicio a otras peticiones.
50.             }
51.          }  // fin while
52.       } finally {
53.          // Liberamos recursos
54.
55.          try {
56.             socket.disconnect();
57.          } catch (Exception ex) {
58.             // Ignoramos las excepciones
59.          }
60.
61.          try {
62.             socket.close();
63.          } catch (Exception ex) {
64.             // Ignoramos las excepciones
65.          }
66.       }
67.    }
68. }
69.
```

Código fuente de ejemplo de integración en un servicio:

```
view plain  print  ?
01. package com.autentia.tutoriales.servicemonitor;
02.
03. /**
04.  * Esta clase no hace más que mostrar como se integraría el StatusListener
05.  * dentro de nuestra lógica de negocio de cada aplicación.
06.  * @author Carlos García. Autentia.
07.  */
08. public class ServiceMonitorApp extends Thread {
09.
10.     private StatusListener status;
11.
12.     /**
13.      * Constructor
14.      */
15.     public ServiceMonitorApp() {
16.         try {
17.             // Escuchará las peticiones de estado en el puerto 5000.
18.             // (Debería ser leido de un archivo de configuración)
19.             this.status  = new StatusListener(5000);
20.         } catch (java.net.SocketException ex){
21.             throw new java.lang.InstantiationError(ex.getMessage());
22.         }
23.
24.         this.start();
25.
26.     }
27.
28.     /*
29.      * @see java.lang.Thread#run()
30.      */
31.     public void run(){
32.         // Activamos la escucha de peticiones de estado.
33.         status.start();
34.
35.         try {
36.             while (true){
37.                 // Esta aplicación no hace nada, solo lanza el Thread de escucha de peticiones de estado
38.                 Thread.sleep(1000);
39.             }
40.         } catch (java.lang.InterruptedException ex){
41.             // El proceso ha sido interrumpido
42.         }
43.
44.         // Detenemos la escucha de peticiones de estado
45.         status.interrupt();
46.     }
47.
48.     /**
49.      * Inicia el servidor de ejemplo con escuchas de estado integrado.
50.      */
51.     public static void main(String[] args) {
52.         new ServiceMonitorApp();
53.     }
54. }
55.
```

La integración se puede observar en las líneas 19, 33 y 45.

Código fuente para consultar el estado del servicio:

```
view plain  print  ?
01. package com.autentia.tutoriales.servicemonitor;
02. import java.net.*;
03.
04.
05. /**
06.  * Testea que el servicio esté funcionando correctamente.
07.  * @author Carlos García. Autentia.
08.  */
09. public class ServiceStatusTest {
10.
11.    /**
12.     * Realiza una consulta de estado al servicio
13.     * @param args Son requeridos los siguientes parámetros: <direccion_ip>, <puerto>
14.     */
15.    public static void main(String[] args) {
16.       DatagramSocket  socket   = null;
17.       DatagramPacket  packet   = null;
18.       DatagramPacket  received = null;
19.       SocketAddress   target   = null;
20.       byte[]          bytes1   = "STATUS".getBytes();
21.       byte[]          bytes2   = new byte[64];
22.       String          receivedData = null;
23.
24.       try {
25.          if (args.length != 2){
26.             System.err.println("Formato: ServiceStatusTest <direccion_ip> <puerto>");
27.             return;
28.          }
29.
30.          // Dirección y puerto del servicio.
31.          target = new InetSocketAddress(args[0], Integer.parseInt(args[1])); // (ip y puerto)
32.
33.          // Creamos el paquete de consulta
34.          packet = new DatagramPacket(bytes1, bytes1.length, target);
35.
36.
37.          // Instanciamos el socket y configuramos un tiempo máximo de 5 segundos para obetener una resp
38.          socket = new DatagramSocket();
39.          socket.setSoTimeout(5000);
40.
41.          // Enviamos la petición
42.          socket.send(packet);
43.
44.          // Leemos la respuesta
45.          received = new DatagramPacket(bytes2, bytes2.length);
46.          socket.receive(received);
47.
48.          receivedData = new String(received.getData(), 0, received.getLength());
49.
50.       } catch (Exception ex){
51.          // No hacemos nada especial.
52.          // (El tratamiento se hace en el finally).
53.       } finally {
54.          // Liberamos recursos
55.          try {
56.             socket.close();
57.          } catch (Exception ex){}
58.
59.          // Devolvemos el código de respuesta
60.          if ("ok".equalsIgnoreCase(receivedData)){
61.             Runtime.getRuntime().exit(0);   // OK
62.          } else {
63.             Runtime.getRuntime().exit(-1);  // KO
64.          }
65.       }
66.    }
67. }
68.
69.
```

A continuación vamos a hacer que el servicio responda a las peticiones: start, stop, restart, status.

En Unix, los servicios suelen responder a estos mensajes (al menos a start y stop). Para ellos cada servicio tiene un script en el directorio `/etc/init.d`.

Script de inicio, parada, reinicio, status. example_service.sh:.

(Lo debemos crear en el directorio `/etc/init.d`)

```
#!/bin/bash

# ----------------------------------------------------------------------
# Este Script para iniciar, parar y monitorizar el servicio de ejemplo
# debe ser colocado en el directorio /etc/init.d
# ----------------------------------------------------------------------

SERVICE_PATH="/home/carlosgarcia"

case "$1" in
        start)
          sh $SERVICE_PATH/example_service_start.sh
          ;;
        stop)
          sh $SERVICE_PATH/example_service_stop.sh
          ;;
        restart)
          sh $SERVICE_PATH/example_service_restart.sh
          ;;
        status)
          sh $SERVICE_PATH/example_service_status.sh
          ;;
        *)
          echo "Uso: $0 {start|stop|restart|status)}"
          exit 1
          ;;
esac

exit 0
```

Reinicia el servicio de ejemplo. example_service_restart.sh:

```
#!/bin/bash
# Este Script reinicia el servicio de ejemplo

SERVICE_PATH="/home/carlosgarcia"

sh $SERVICE_PATH/example_service_stop.sh
sh $SERVICE_PATH/example_service_start.sh
```

Inicia el servicio de ejemplo para este tutorial. example_service_start.sh:

```
#!/bin/bash
# Este Script para inicia el servicio de ejemplo para este tutorial

SERVICE_PATH="/home/carlosgarcia"
PID_FILE="$SERVICE_PATH/example_service.pid"

cd $SERVICE_PATH

# Iniciamos el servicio de ejemplo
$JAVA_HOME/bin/java -cp $SERVICE_PATH com.autentia.tutoriales.servicemonitor.ServiceMonitorApp &

# Guardamos el PID del proceso en un archivo
echo $! > "$PID_FILE"
```

Detiene el servicio de ejemplo para este tutorial. example_service_stop.sh:

```
#!/bin/bash
# Este Script detiene el servicio de ejemplo

SERVICE_PATH="/home/carlosgarcia"
PID_FILE="$SERVICE_PATH/example_service.pid"

# Cargamos en una variable el PID del proceso que estaba almacenado en un archivo.
read  PID_VAR < $PID_FILE

# Detenemos el proceso
kill $PID_VAR

# Eliminamos el archivo en donde almacenábamos el PID del proceso.
rm -f $PID_FILE
```

Consulta el estado del servicio de ejemplo para este tutorial. example_service_status.sh:

```
#!/bin/bash
# Este Script consulta el estado del servicio de ejemplo

SERVICE_PATH="/home/carlosgarcia"

$JAVA_HOME/bin/java -cp $SERVICE_PATH com.autentia.tutoriales.servicemonitor.ServiceStatusTest

if test $? -eq 0
then
        echo "Estado: OK"
else
        echo "Estado: KO"
fi
```

A continuación mostramos las comandos que deberíamos escribir para iniciar, detener, reiniciar y consultar el estado del servicio.

- /etc/init.d/example_service start
- /etc/init.d/example_service stop
- /etc/init.d/example_service restart
- /etc/init.d/example_service status

Monitorización del servicio para iniciarlo automáticamente en caso de que esté detenido.

Inicia el servicio en caso de que no responda a la consulta de STATUS example_service_monitor.sh:

```
#!/bin/bash

# Este Script inicia el servicio en caso de que no responda a la consulta de STATUS

SERVICE_PATH="/home/carlosgarcia"

$JAVA_HOME/bin/java -cp $SERVICE_PATH com.autentia.tutoriales.servicemonitor.ServiceStatusTest

# Consultamos el valor de retorno. (
if test $? -ne 0
then
  sh $SERVICE_PATH/example_service_restart.sh
fi
```

(Observe las líneas 61 y 63 del cliente ServiceStatusTest)

Ahora tecleamos las dos líneas siguientes para monitorizar el estado del servicio

Vamos a apoyarnos en el cron (gestor de tarea programadas) para que monitorize el estado del servicio cada 5 segundos

- (La siguiente línea abrirá un editor para editar el archivo crontab del usuario)
  crontab -e

- (Debemos introducir esa linea para monitorizar el estado cada 5 segundos)
  */5 * * * * "sh /home/carlosgarcia/example_service_monitor.sh"

Ojo!! En algunas versiones de `cron` no funciona */5, pruebe con un 5.

Iniciando el servicio en el arranque de la máquina

Si deseamos que el servicio de ejemplo se inicie automáticamente cuando la máquina arranque debemos seguir los siguientes pasos:

- Copiamos o movemos el archivo `example_service` en el directorio `/etc/init.d` (Si no lo habÃamos hecho ya).

- (Creamos un enlaze simbólico en el nivel de arranque 2.

  ```
  ln -s /etc/init.d/example_service /etc/rc2.d/S99example_service.
  ```

  Dentro del sistema operativo, existen varios niveles (de 0 a N) y dentro de cada nivel los servicios se inician según un orden (0-99).
  La S significa "START".

  Por ejemplo, dentro del nivel 2 (directorio /etc/rc2.d) un servicio de nombre S59Servicio1 se iniciará antes que un sercivio con nombre S70Servicio2 (del mismo nivel), pero después de un servicio de nombre S88Servicio3 de el nivel 1. (directorio /etc/rc1.d)

Un saludo.

Carlos García. Autentia.

SOME RIGHTS RESERVED This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 2.5 License.

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

Gestión de contenidos

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	
	Enviar

Otros Tutoriales Recomendados (También ver todos)

Nombre Corto	**Descripción**
Backup Manager: Herramienta de Backup en sistema linux	En este tutorial se explica el funcionamiento de Backup Manager, una herramienta muy completa y sencilla para la realización de copias de seguridad de sistema de ficheros locales y remotos, bases de datos, etc...
Backup en Linux	En este tutorial os mostramos cómo realizar copias de seguridad en un sistema Linux, utilizando sencillas herramientas del propio sistema operativo

Activación automatica de servicios RMI	En este tutorial veremos cómo utilizar el servicio de activación de objetos RMI directamente desde el cliente de forma automática
Desarrollo Gráfico Scripts Ant	Os mostramos como crear y ejecutar scripts Ant (para automatizar tareas en el mundo Java) con las herramientas gratuitas Antelope y NetBeans
Subdomio Gratis en NoIP	Si no tienes una IP fija y quieres tener un servidor en casa, necesitas que la gente acceda a ti a través de un alias (nombre de domino). Si tus direcciones (URLs) son largas o complicadas puedes crearte alias más sencillos.
Emular Ubuntu en XP con QEMU	Os mostramos como instalar y configurar Qemu en XP al mismo tiempo que montar la consola QemuManager e instalar sobre Qemu el sistema operativo Ubuntu (tipo Linux Debian) desde una imagen del cd

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 Opimizado 800X600