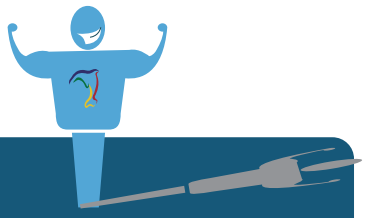
autentia

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...

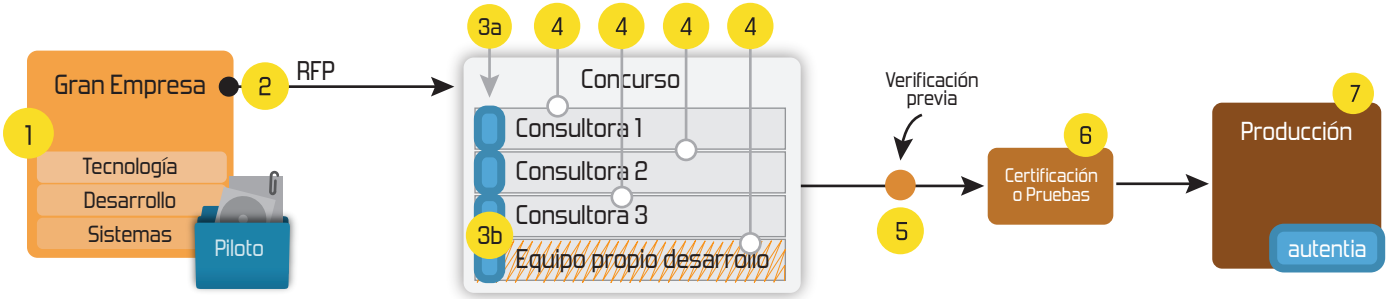
1. Desarrollo de componentes y proyectos a medida



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

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: 2010-07-22

Creación de servicios web RESTful con el soporte de RESTeasy de Jboss Seam.

0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno.
- 3. Configuración.
- 4. Ejemplo práctico.
- 5. Referencias.
- 6. Conclusiones.

1. Introducción

Jboss Seam incluye entre sus módulos el proyecto RESTeasy, una implementación de la especificación JAX-RS (JSR 311), cuya implementación de referencia es Jersey. RESTeasy es también un proyecto de Jboss y al incluirse dentro de Seam vamos a tener el soporte para la generación de servicios web "ligeros", sin necesidad de incluir en los mensajes de comunicación la envoltura SOAP.

Al integrarse dentro de Jboss Seam nos va a permitir convertir un componente de Seam (un POJO o un EJB) en un web service, con lo que se beneficia de las características propias de un componente Seam: manejado por el contenedor, inyección de dependencias, eventos sobre el ciclo de vida, gestión declarativa de transacciones,..

Este tutorial muestra cómo configurar un proyecto Seam para dar soporte de RESTeasy y un ejemplo de publicación de un servicio web.

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
- Jboss Seam 2.2.0.GA
- Maven 2.2.1.
- Eclipse 3.5: Galileo, con IAM (plugin para Maven).
- GlassFish v.2.1 con la jdk 1.6.

3. Configuración.

RESTeasy es un módulo de Seam que no se distribuye con las librerías del core, con lo que lo primero que debemos hacer es incluir la dependencia en el módulo web que lo necesite:

```
<DEPENDENCY>
    <GROUPID>org.jboss.seam</GROUPID>
    <ARTIFACTID>jboss-seam-resteasy</ARTIFACTID>
    <VERSION>${seam.version}</VERSION>
    <EXCLUSIONS>
        <EXCLUSION>
            <GROUPID>org.jboss.seam</GROUPID>
            <ARTIFACTID>jboss-seam</ARTIFACTID>
        </EXCLUSION>
    </EXCLUSIONS>
</DEPENDENCY>
```

En nuestro caso ${seam.version} es una propiedad que tenemos definida a nivel de pom.xml para su reutilización y llevamos a cabo una exclusión de la librería del core puesto que la dependencia la arrastramos del módulo padre y al empaquetarse como un ear, va a ese nivel y no a nivel de war.

Lo siguiente es revisar la configuración del servlet de recursos de Jboss Seam a nivel del descriptor de despliegue del proyecto web (web.xml), es quien va a dar soporte a las peticiones a través de http a los servicios:

```
<servlet>
    <servlet-name>Seam Resource Servlet</servlet-name>
    <servlet-class>org.jboss.seam.servlet.ResourceServlet</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>Seam Resource Servlet</servlet-name>
    <url-pattern>/resources/*</url-pattern>
</servlet-mapping>
```

Por último debemos añadir lo siguiente al fichero de configuración de Seam (components.xml):

```
<resteasy:application resource-path-prefix="/ws" />
```

Se trata del path relativo al path de recursos que va a dar servicio a los servicios web.

De este modo, la invocación a los servicios se realizará a través de la siguiente url: **http://localhost:8080/app/resources/ws/servicio**, donde localhost es la máquina, app es el contexto de la aplicación y servicio el nombre del servicio.

Podemos, además, configurar un nivel de seguridad de modo que la petición de un servicio solicite la autenticación del usuario a través de login y contraseña incluyendo lo siguiente en el mismo fichero components.xml:

```
<web:authentication-filter
        url-pattern="/resources/ws/*"
        auth-type="basic"
        realm="PLV Web Service layer" />
```

Por defecto enlazará con la configuración de seguridad definida en el mismo fichero vía <security:identity authenticate-method=

4. Ejemplo práctico.

A continuación mostramos un ejemplo de un componente anotado para su publicación con un web service con el soporte de REST. La idea es la de disponer de un servicio en la red que permita marcar ordenes de compra (PurchaseOrder) como cerradas y su invocación la pueda realizar cualquier cliente sin necesidad de hablar en SOAP..

```
package com.autentia.web.ws;

import java.io.Serializable;
```

EGit un plugin de Eclipse para el sistema de control de versiones distribuido Git

Crear una estructura compleja del tipo Array en un servicio web Axis2

Apache TCPMON : El Sniffer de los Servicios Web

Desplegando en Tomcat un proyecto web con Hudson.

Últimos Tutoriales del Autor

Facelets en JSF 2: sistema de plantillas y componentes por composición.

DbVisualizer free version.

Session TimeOut en RichFaces, con el soporte de Jboss Seam.

Retrasar la carga de Javascript con jQuery.getScript().

Optimización de páginas web con Page Speed.

Síguenos a través de:

Últimas ofertas de empleo

2010-06-25
T. Información - Analista / Programador - BARCELONA.

```
04
05 import javax.ws.rs.GET;
06 import javax.ws.rs.Path;
07 import javax.ws.rs.PathParam;
08 import javax.ws.rs.Produces;
09
10 import org.jboss.seam.annotations.In;
11 import org.jboss.seam.annotations.Logger;
12 import org.jboss.seam.annotations.Name;
13 import org.jboss.seam.annotations.security.Restrict;
14 import org.jboss.seam.log.Log;
15
16 import com.autentia.core.services.PurchaseOrderManager;
17
18 /**
19  * Prueba de concepto de la publicación de un WS vía RESTful.
20  *
21  * @author Autentia Real Business Solutions S.L.
22  * @see http://www.autentia.com
23  */
24 @Name("purchaseOrderResource")
25 @Path("/purchaseOrder")
26 @Transactional
27 public class PurchaseOrderResource implements Serializable {
28
29     private static final long serialVersionUID = 8655977096519941447L;
30
31     @Logger
32     private Log log;
33
34     @In
35     private PurchaseOrderManager purchaseOrderManager;
36
37     @GET
38     @Path("/{reference}")
39     @Produces("text/plain")
40     @Restrict("#{s:hasRole('ADMIN')}")
41     public String closePurchaseOrder(@PathParam("reference") String reference) {
42         log.debug("request received [reference:'" + reference + "']");
43         purchaseOrderManager.closePurchaseOrder(reference);
44         return "ok";
45     }
46
47 }
```

Para invocarlo debemos realizar una petición a la siguiente url: **http://localhost:8080/app/resources/ws/purchaseOrder/80002311**, donde 80002311 es el número de referencia de la orden de compra a cerrar.

Vamos a repasar las anotaciones:

- 24: @Name("purchaseOrderResource") publica la clase como un componente de Seam con el nombre purchaseOrderResource.
- 25: @Path("/purchaseOrder") publica el componente como un servicio web accesible a través del path relativo purchaseOrder. Seam escanea las clases en busca de esta anotación de modo que no se requiere más configuración que esta.
- 26: @Transactional marca el ámbito de las llamadas a los puntos finales del servicio como transaccionales, si en vez de ser un POJO se tratase de un EJB no sería necesario puesto que es su comportamiento por defecto.
- 31: @Logger asigna una instancia del objeto de trazas manejado por el contenedor, de modo que no es necesario invocar a un método estático de una factoría para obtenerlo, nos viene inyectado (es una facilidad propia de Seam).
- 34: @In inyectamos la dependencia de un servicio (otro componente de Seam) que contiene la lógica de negocio necesaria para marcar como cerrada una orden de compra en función de su identificador.
- 37: @GET marcamos el método como accesible a través del método GET, una petición GET vía http.
- 38: @Path("/{reference}") publicamos el método como punto final del servicio web accesible a través de un path relativo al servicio que declara un único parámetro y lo marca con el nombre de variable "reference".
- 39: @Produces("text/plain") indica el tipo de respuesta del servicio, para esta prueba sencilla devolvemos una cadena.
- 40: @Restrict("#{s:hasRole('ADMIN')}") es una anotación propia de Seam que permite restringir la invocación al método únicamente a usuarios autenticados y con un nivel de autorización predeterminado.
- 41: @PathParam("reference") asigna el valor de la variable marcada con el nombre reference en la anotación de la línea 38 al parámetro de entrada del método.

Con todo ello, en la línea 43 hacemos uso del servicio que nos viene inyectado invocando a un método del mismo que recibe el número de referencia que obtenermos directamente de la url que solicita el servicio.

5. Referencias.

- http://docs.jboss.org/seam/2.2.0.GA/reference/en-US/html/webservices.html

6. Conclusiones.

Jboss Seam nos da soporte a múltiples tecnologías de integración, en este tutorial hemos visto un ejemplo de una de ellas, que nos permite exponer lógica de negocio de una manera sencilla hacía el exterior.

Si estáis interesados en el contenido de nuestros tutoriales y tenéis una necesidad formativa al respecto no dudeis en poneros en contacto con nosotros. En Autentia nos dedicamos, además de a la consultoría, desarrollo y soporte a desarrollo, a impartir cursos de formación de las tecnologías con las que trabajamos.

Un saludo.

Jose

jmsanchez@autentia.com