

[Inicio](#)[Quiénes somos](#)[Formación](#)[Publicaciones](#)[Tutoriales.](#)[adictosaltrabajo](#) / [Tutoriales](#) / [Tutorial de BPEL con OpenESB \(II\)](#)

Iván García Puebla

Consultor tecnológico de desarrollo de proyectos informáticos.



Tutorial de BPEL con OpenESB (II)

diciembre 29, 2009

Iván García Puebla

4 Comentarios



Tutoriales

17302 visitas

Tutorial de BPEL con OpenESB (II)

Indice

1. Tutorial de BPEL con OpenESB (I)

1. Introducción
2. El proceso de negocio
3. Identificar los servicios web requeridos
4. Modelar el proceso de negocio con BPEL
 1. Generar el descriptor del proceso BPEL
 2. Importar descriptores WSDL de los servicios involucrados
 3. Diseñar el proceso BPEL gráficamente

2. Tutorial de BPEL con OpenESB (II)

1. **Desplegar el módulo BPEL en el ESB**
2. **Probar el proceso de compra**
3. **Conclusión**

Tutorial de BPEL con OpenESB (II)

Esta es la segunda parte del tutorial. La primera se encuentra en [Tutorial de BPEL con OpenESB \(I\)](#).

Desplegar el módulo BPEL en el ESB

En OpenESB, todo módulo de la categoría SOA será desplegado en el contenedor de JBI dentro de una Composite Application. Por ello creamos en NetBeans un proyecto de este tipo desde el menú File | New Project | SOA | Composite Application y lo nombramos AplicacionTuLibroOnline:

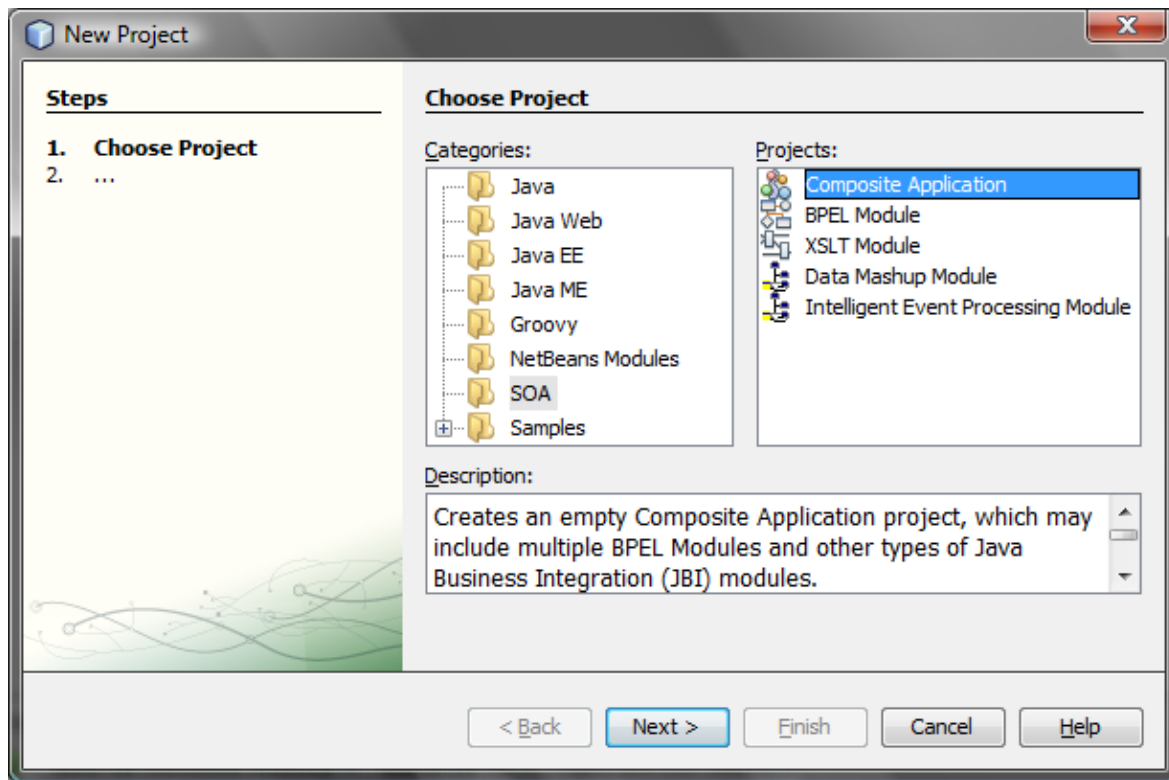


Los más visitados de la semana

[Primeros pasos con Gulp](#)

[Validación en lado cliente con HTML5](#)

[Un entorno de programación en la nube: un vistazo a Cloud 9 ...](#)



Ejemplo de uso con JSF 2.0,
Primefaces e Hibernate
Tutorial básico de bases de datos
en Java mediante JDBC

Tweets por @adictosaltrabaj



adictosaltrabajo
@adictosaltrabaj

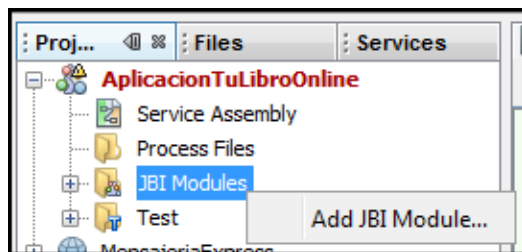
TUTORIAL | Un entorno de programación en la nube. Un vistazo a #Cloud9 IDE, por @Paniadri
adictosaltrabajo.com/tutoriales/un-...



Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies, pinche el enlace para mayor información

ACEPTAR



Añadir módulo SOA

seleccionamos el proyecto ServiciosVenta y pulsamos el botón Add Project JAR Files:

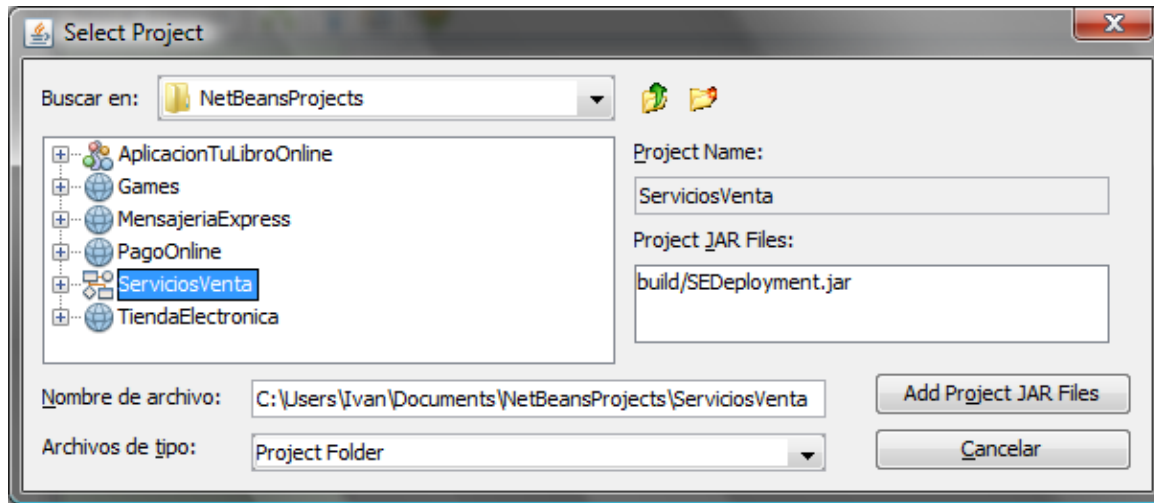
Insertar

Ver en Twitter

Archivos

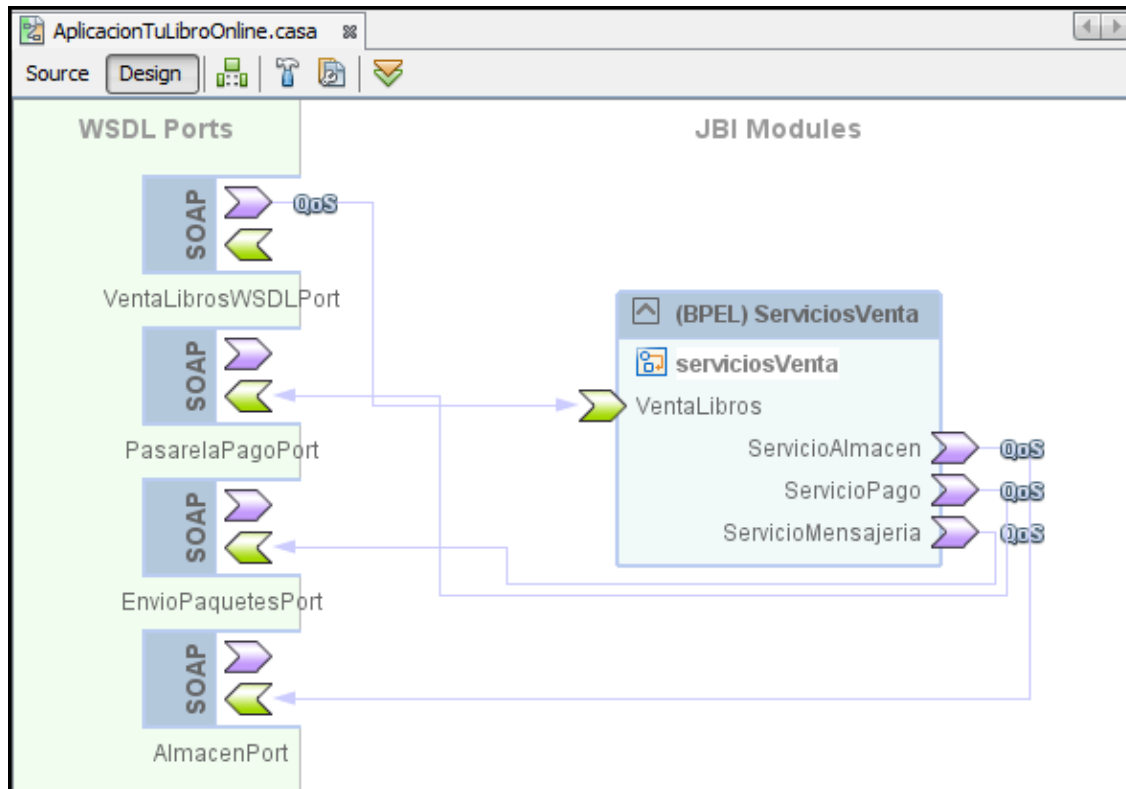
Archivos

Elegir mes



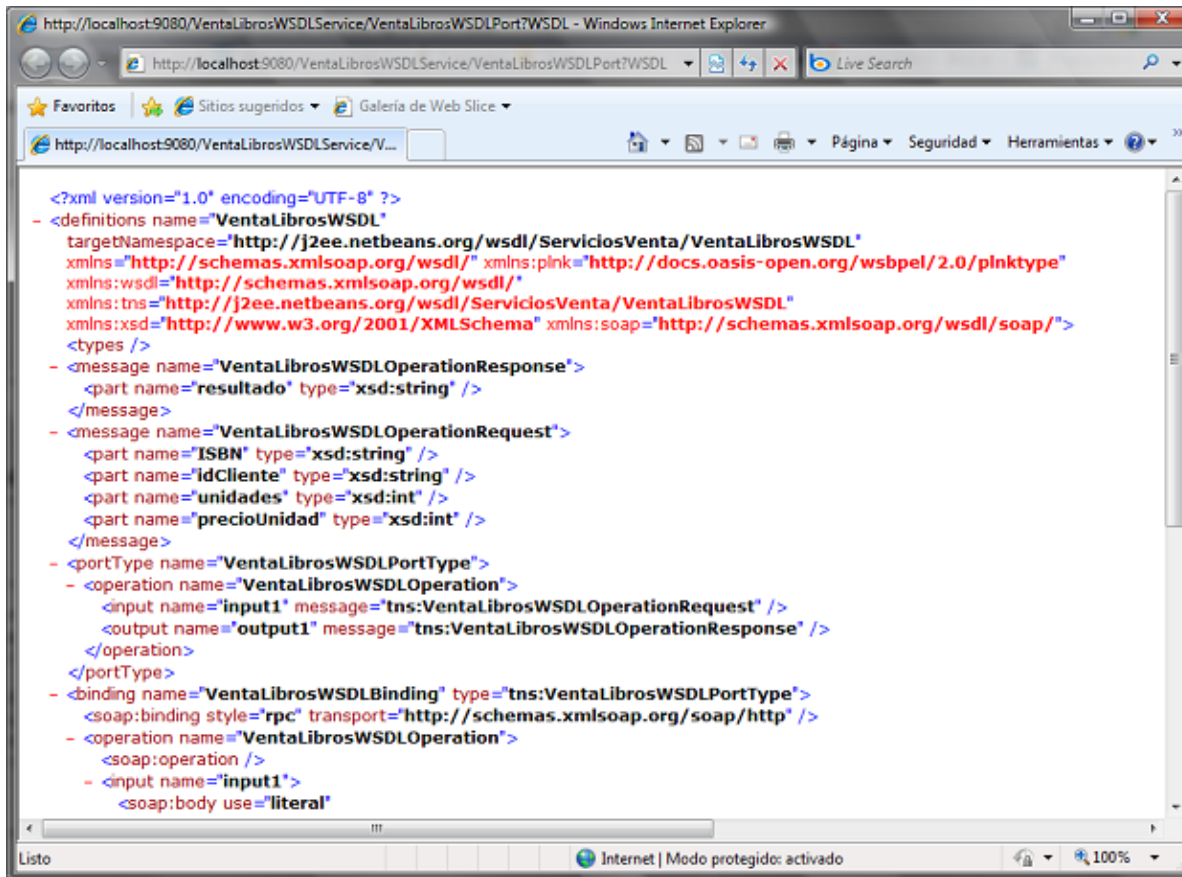
Seleccionamos nuestro proyecto BPEL

A continuación compilamos los proyectos introducidos seleccionando Clean and Build haciendo botón derecho sobre el nombre del proyecto, AplicacionTuLibroOnline. El resultado se mostrará en el editor CASA (Composite Application Service Assembly):



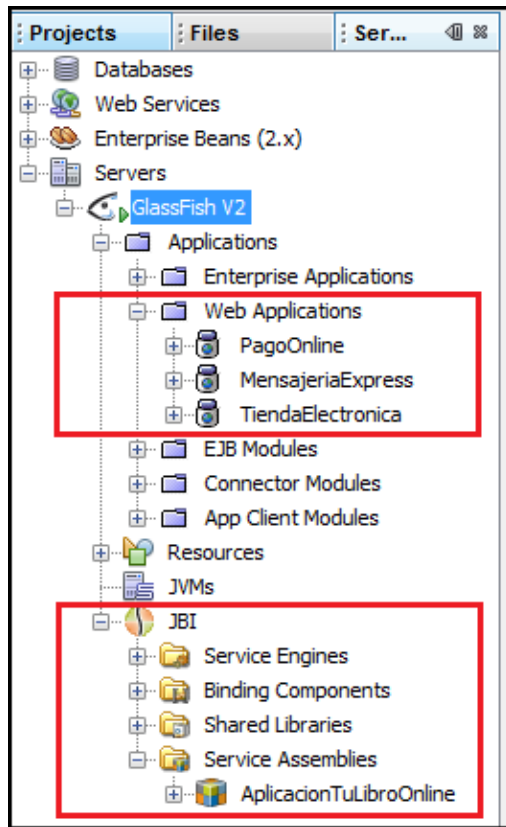
Editor de OpenESB para composición de aplicaciones (CASA)

Guardamos el proyecto y lo desplegamos, haciendo de nuevo botón derecho sobre el nombre del proyecto | Deploy. El proceso BPEL será ahora un web service publicado en: <http://localhost:9080/VentaLibrosWSDLService/VentaLibrosWSDLPort?WSDL>



WSDL del web service implementado con BPEL

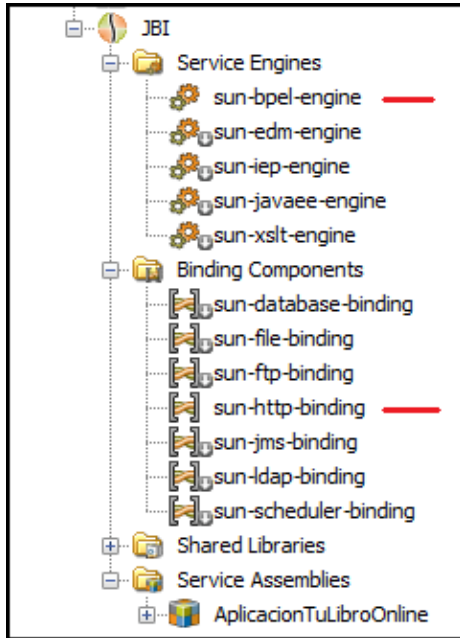
Resulta interesante ver el despliegue de los distintos artefactos que estamos utilizando. En la sección Services en el panel izquierdo de NetBeans vemos que las aplicaciones web que contienen los web services están desplegadas en el contenedor de aplicaciones web de GlassFish, y que la aplicación compuesta SOA está desplegada dentro del contenedor de JBI (Java Business Integration):



Artefactos desplegados en OpenESB

Nuestra aplicación BPEL se apoya en las siguientes dos funcionalidades del ESB (implementadas como módulos JBI):

- Conectividad a servicios por HTTP: HTTP Binding Component (sun-http-binding)
- Motor de ejecución del código fuente BPEL que hemos editado gráficamente: BPEL Service Engine (sun-bpel-engine)

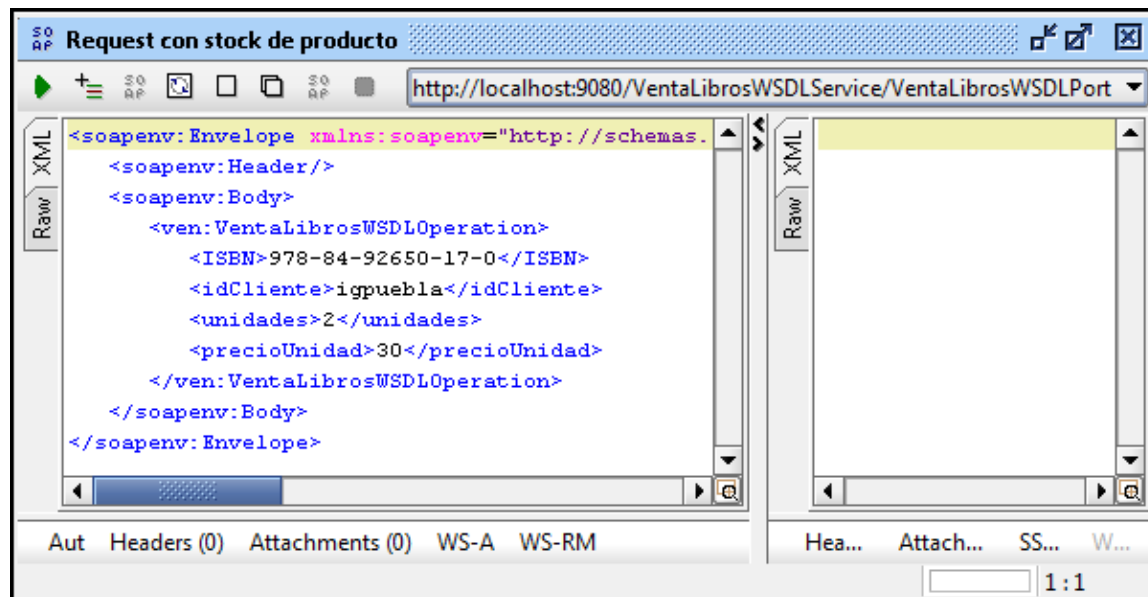


JBI, el core del ESB de OpenESB

Probar el proceso de compra

Podemos enviar mensajes SOAP sobre HTTP al servicio desarrollando un cliente o simulándolo con alguna aplicación como soapUI. También es posible generar test unitarios con junit desde el propio proyecto AplicacionTuLibroOnline, carpeta Test (véase documentación de OpenESB: [Testing The Composite Application](#))

Utilizando soapUI (tutorial: [soapUI: jugando con web services](#)) creamos el mensaje de SOAP request al servicio web con los datos para comprar un libro:



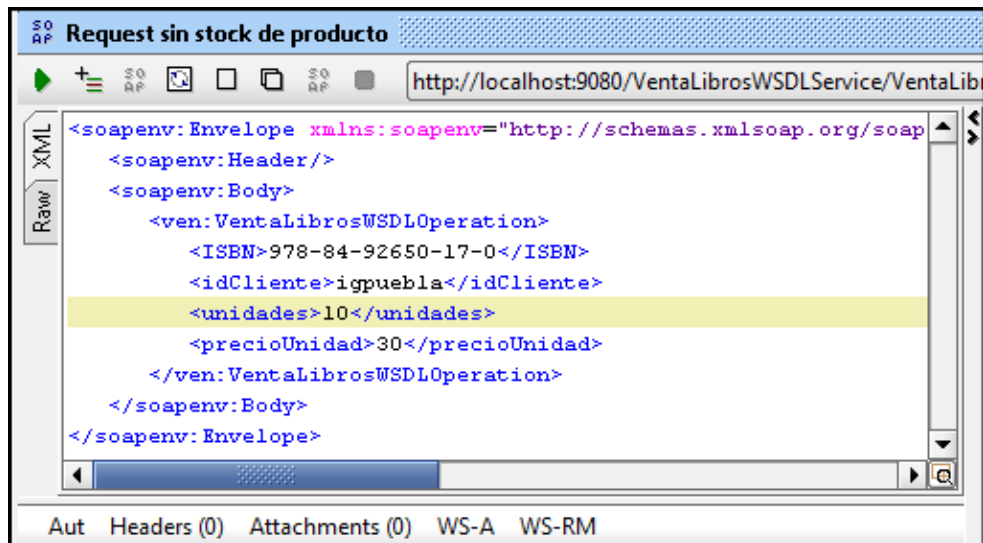
Mensaje SOAP request de compra

El servicio nos responde con la información de la compra efectuada:



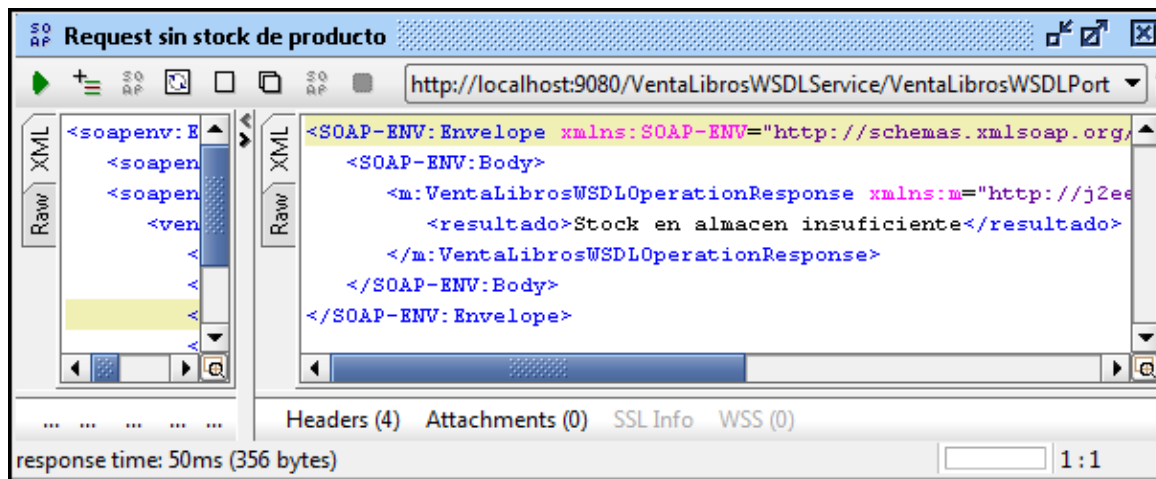
Mensaje SOAP response con información de la compra

Hagamos otra prueba solicitando 10 unidades del producto (aunque el almacén no sirve pedidos de más de 5 libros):



Compra de 10 unidades de un libro

El resultado será un mensaje de notificación de la falta de stock para el pedido.



La editorial no puede servir un pedido tan grande...

A la vista de estos resultados, el proceso de negocio ha sido implementado en BPEL con éxito.

Conclusión

BPEL (aka BPEL4WS o *BPEL for Web Services*) es un estándar para la orquestación de servicios web. Su código fuente está basado en XML y extensiones al lenguaje XPath, y puede ser ejecutado en motores que habitualmente están incluidos en implementaciones ESB de libre uso o comerciales.

Como hemos visto, OpenESB ofrece en NetBeans un editor gráfico que permite modelar fácilmente procesos de negocio y en GlassFish ESB el soporte para su ejecución. Extrapolando el ejemplo del tutorial a procesos de negocio reales, podemos hacernos una idea del potencial que ofrecen este tipo de tecnologías para los entornos empresariales en los que tales procesos cambian con frecuencia según sus estrategias de mercado.

Comparte este artículo!



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Respuestas (4)



Marcelo Palma

octubre 30, 2014 at 11:45 pm · Responder

Te felicito por el ejemplo, me ayuda mucha a comprender BPEL, ahora estoy implementando un Web

Services con WS-Security!

Gracias



Lucio Marcelo Quispe Ortega

octubre 30, 2014 at 11:46 pm · Responder

Hola Iván, excelente tu trabajo, sin embargo no pude hacerlo correr en Netbeans 6.5.1, por favor si tuvieras alguna solución, te estaré agradecido. El error es el siguiente:

Deploying a service assembly...

host=localhost

port=4848

file=D:\\\\Mis

documentos\\\\NetBeansProjects\\\\AplicacionTuLibroOnline/dist/AplicacionTuLibroOnline.zip

ERROR: Successful execution of Deploy: D:\\\\Mis

documentos\\\\NetBeansProjects\\\\AplicacionTuLibroOnline/dist/AplicacionTuLibroOnline.zip

WARNING: (JBIMA0404) La implementación del conjunto de servicios AplicacionTuLibroOnline se ha realizado parcialmente; no se han podido implementar algunas unidades de servicio.

* Component: sun-http-binding

ERROR: (SOAPBC_DEPLOY_2) HTTPBC-E00201: Error en la implementación.

javax.wsdl.WSDLException: WSDLException (at /definitions/types/xsd:schema):

faultCode=OTHER_ERROR: An error occurred trying to resolve schema referenced at

\\\\'http://localhost:8080/TiendaElectronica/AlmacenService?xsd=1\\\\', relative to

\\\\'file:/C:/Archivos de programa/glassfish-v2.1/domains/domain1/jbi/service-

assemblies/AplicacionTuLibroOnline.2/AplicacionTuLibroOnline-sun-http-binding/sun-http-

binding/META-

INF/ServiciosVenta/src/ServicioAlmacen/localhost_8080/TiendaElectronica/AlmacenService.wsdl\\\\':

java.net.URISyntaxException: Illegal character in path at index 17: file:/C:/Archivos de

programa/glassfish-v2.1/domains/domain1/jbi/service-

assemblies/AplicacionTuLibroOnline.2/AplicacionTuLibroOnline-sun-http-binding/sun-http-

binding/META-

INF/ServiciosVenta/src/ServicioAlmacen/localhost_8080/TiendaElectronica/AlmacenService.wsdl

D:\\\\Mis documentos\\\\NetBeansProjects\\\\AplicacionTuLibroOnline\\\\

bproject\\\\build-impl.xml:213: Deployment failure.

BUILD FAILED (total time: 3 seconds)

Nota.- Cabe aclarar que los servicios corren bien, el problema se da cuando le doy deploy a AplicacionTuLibroOnline. Tengo activados los componentes sun-bpel-engine y sun-http-binding.



gaston marquez

octubre 30, 2014 at 11:46 pm · Responder

Hola Iván, junto con felicitarte y agradecerte este tutorial, tengo el mismo problema que \\\"marceloqo\\\" ¿como se podría solucionar ese detalle?

Saludos!



Oscar José Antonio Pinto Salazar

octubre 30, 2014 at 11:46 pm · Responder

Estimado, excelente, tremendo aporte..... un muy buen ejemplo..

Deja un comentario

Lo último

Pruebas multiplataforma con Selenium GRID

Un entorno de programación en la nube: un vistazo a Cloud 9 IDE

Primeros pasos con Gulp

Primeros pasos con Kubernetes

Instalación de Kubernetes en

Ubuntu con Ansible

Datos de contacto

Edificio BestPoint Avd. de Castilla, 1, Planta 2, Oficina 21B (San Fernando de Henares)

Phone: 916 75 33 06

E-Mail: info@adictosaltrabajo.com

Web: <http://www.autentia.com>

Powered by



Copyright 2003-2016 © All Rights Reserved | Texto legal y condiciones de
uso