

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

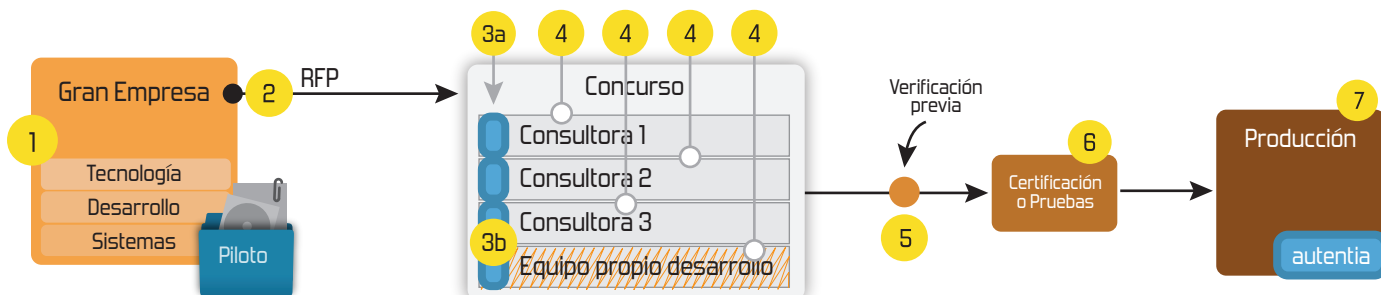
1. Desarrollo de componentes y proyectos a medida



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](#)

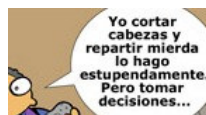
[Libro de Visitas](#)

[Comic](#)

NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha **aquí...**

[Ver cursos que ofrece Autentia](#)

[Descargar comics en PDF y alta resolución](#)



[iNUEVO!] 2008-04-01



2008-03-25



2008-03-17



2008-03-11

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

Fecha de creación del tutorial: 2008-04-03

¿?

Servicios Web RESTful en Axis 2

En este tutorial vamos a realizar una descripción de REST y vamos a ver un ejemplo práctico de un Servicio Web RESTful

¿Qué es REST?

Representational Estado Transfer (REST) es un método de diseño de arquitectura que consiste en ver un servicio Web como un recurso identificable mediante una URL.

¿Cuándo usar REST?

Un diseño REST puede ser apropiado cuando:

1. Cuando la funcionalidad expuesta por el servicio es síncrona.
2. Cuando el servicio Web no tiene estado.
3. Cuando la información que provee el servicio puede ser cacheada para mejorar el rendimiento.
4. Cuando se conoce bien el contexto de la comunicación (formato de datos, métodos, etc.) entre productor y el consumidor.
5. Cuando es importante que la comunicación sea ligera en términos de bytes transmitidos debido al coste » PDAs, móviles, etc
6. Cuando el consumidor tiene limitaciones de ancho de banda (PDS, móviles, etc).
7. Cuando el consumidor desea acceder a los servicios (sin tener que instalar nada), a través de una página Web con Ajax y analizar el resultado (xml).

¿Qué son los Web Services RESTful?

Son aquellos servicios Web que funcionan bajo REST

Los Web Service REST proporcionan acceso a través de los métodos GET y POST de HTTP.

1. **GET:** En accesos vía GET, tanto las operaciones como los parámetros se pasan por la URL » Sólo soporta argumentos con tipos simples.
2. **POST:** En este tipo de acceso, la información no viaja en mensajes SOAP Envelope, sino directamente en el payload del mensaje.

REST en Axis2

Axis2 puede ser configurado para funcionar con REST, de manera que pueda enviar y recibir peticiones REST. (Por defecto está configurada esta habilitada)

[Anuncios Google](#)

[Java Web Service](#)

[Certificación Java](#)

[Java Rest](#)

[Visual Basic C#](#)

[Apache Axis Soap](#)

Catálogo de servicios Autentia (PDF 6,2MB)



[En formato comic...](#)

Google™

☐ Web

☒ [www.adictosaltrabajo.com](#)

Últimos tutoriales

2008-04-05

[JMX y monitorización de JBoss](#)

2008-04-05

[Jersey: la implemetación de RESTful de Sun](#)

2008-04-05

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

2008-04-05

[Metro: pila de webservices de Sun.](#)

2008-04-04

[Espectaculares efectos visuales en el escritorio de Linux, con Compiz Fusion](#)

2008-04-04

[Monitorización de Web Services con Glassfish Wsmonitor](#)

2008-04-04

[Axis2. Ejemplo de creación de un servicio Web](#)

2008-04-03

[Servicios Web RESTful en Axis 2](#)

2008-04-03

[XML Signature - Firma Digital sobre XML](#)

2008-04-03

[XML Encryption, Criptografía sobre XML](#)

Para habilitar REST, hay que añadir la siguiente línea en el archivo de configuración `axis2.xml`.

```
<parameter name="enableREST" locked="false">true</parameter>
```

Cuando un mensaje es recibido, entonces:

- Si el content type es text/xml y además NO está presente la cabecera SOAP Action » Mensaje REST.
- Si el content type es text/xml y además SI está presente la cabecera SOAP Action » Mensaje SOAP.

Instalación de Axis 2

Para instalar Axis 2, simplemente deberás

1. Descargártelo desde <http://ws.apache.org/axis2/>
2. Descomprimirlo en tu disco duro
3. Crear una variable de entorno de nombre `AXIS_HOME` que apunte al directorio donde fue descargado. Por ejemplo
`c:/java/axis2-1.3`
4. Desplegar el archivo `%AXIS_HOME%/dist/axis2.war` en tu servidor de aplicaciones o contenedor de servlets favorito (JBoss, iPlanet, WebSphere, Weblogic, GlassFish, Geronimo, Tomcat, etc.)

A continuación veremos varias maneras de invocar uno de los servicios Web que vienen preinstalados en la distribución de Axis 2

Ejemplo: Invocación del servicio Web a través de HTTP GET

Abra su navegador favorito y ponga la siguiente dirección: <http://127.0.0.1:8080/axis2/services/Version/getVersion>

Ejemplo: Invocación del servicio Web a través de HTTP POST desde un formulario

```
view plain print ?
01. <html>
02. <body>
03. <form name="n1" method="POST" action="http://127.0.0.1:8080/axis2/services/Version/getVersion" >
04. <input type="submit" value="Invocar el Servicio Web" >
05. </form>
06. </body>
07. </html>
```

Ejemplo: Invocación del servicio Web a través de HTTP POST usando las clases de Axis

```
view plain print ?
01. package com.autentia.ws.rest.sample1;
02.
03. import org.apache.axiom.om.OMElement;
04. import org.apache.axis2.addressing.EndpointReference;
05. import org.apache.axis2.client.ServiceClient;
06.
07. /**
08.  * Invocación via POST del Web service
09.  * @author Autentia.
10.  */
11. public class RESTClient {
12.     public static void main(String[] args) throws Exception {
13.         // Instanciamos el cliente
14.         ServiceClient cliente = new ServiceClient();
15.         cliente.setTargetEPR(new EndpointReference("http://127.0.0.1:8080/axis2/services/Version/getVersion")
16.
17.         // Invocamos el servicio Web
18.         OMElement result = cliente.sendReceive(null);
19.
20.         // Imprimimos el resultado por la salida estándar
21.         result.serialize(System.out);
22.     }
23. }
24.
```

Ejemplo: Invocación del servicio Web a través de HTTP POST desde las clases estándares de Java

Últimas ofertas de empleo

2008-04-04
[Banca - Genérico - MADRID.](#)

2008-04-03
[Banca - Genérico - MADRID.](#)

2008-04-02
[T. Información - Analista / Programador - MADRID.](#)

2008-04-02
[T. Información - Analista / Programador - MADRID.](#)

2008-03-29
[T. Información - Analista / Programador - MADRID.](#)

Anuncios Google

view plain print ?

```
01. package com.autentia.ws.rest.sample1;
02.
03. import java.io.*;
04. import java.net.URL;
05. import java.net.HttpURLConnection;
06. import java.net.URLEncoder;
07.
08.
09. /**
10.  * Invocación via POST del Web service
11.  * @author Carlos García. Autentia.
12.  * @see http://www.mobiletest.es
13.  */
14. public class RESTClient2 {
15.     public static void main(String[] args) throws Exception {
16.         // URL del servicio Web
17.         URL url = new URL("http://127.0.0.1:8080/axis2/services/Version/getVersion" );
18.
19.         // Establecemos la conexión
20.         HttpURLConnection connection = (HttpURLConnection) url.openConnection();
21.
22.         // Enviamos los datos asociados a la petición
23.         connection.setDoOutput(true);
24.         OutputStreamWriter wr = new OutputStreamWriter(connection.getOutputStream());
25.         wr.write(URLEncoder.encode( "parametro1", "UTF-8") + "=" + URLEncoder.encode( "valor1", "UTF-8"));
26.         wr.write("&");
27.         wr.write(URLEncoder.encode( "parametro2", "UTF-8") + "=" + URLEncoder.encode( "valor2", "UTF-8"));
28.         wr.flush();
29.
30.
31.         if (connection.getResponseCode() == HttpURLConnection.HTTP_OK){
32.             // Leemos el contenido de la respuesta y lo mostramos por la salida estándar
33.             BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));
34.             String line = reader.readLine();
35.             while (line != null) {
36.                 System.out.println(line);
37.
38.                 line = reader.readLine();
39.             }
40.             reader.close();
41.         } else {
42.             System.out.println(connection.getResponseMessage());
43.         }
44.
45.         // Cerramos la conexión
46.         connection.disconnect();
47.     }
48. }
49.
50.
```

Ejemplo: Invocación del servicio Web a través de HTTP POST usando HTTPClient

Puedes ver un tutorial de HTTPClient [aquí](#)

```

view plain print ?
01. package com.autentia.ws.rest.sample1;
02.
03. import org.apache.commons.httpclient.*;
04. import org.apache.commons.httpclient.methods.*;
05. import org.apache.commons.httpclient.params.HttpMethodParams;
06.
07. import java.io.*;
08.
09. /**
10.  * Invocación via POST del Web service usando HttpClient
11.  * @author Carlos García. Autentia.
12.  * @see http://www.mobiletest.es
13.  */
14. public class RESTClient3 {
15.     /**
16.      * Punto de inicio de ejecución del ejemplo.
17.      */
18.     public static void main(String[] args) {
19.         HttpClient httpClient = null; // Objeto a través del cual realizamos las peticiones
20.         HttpMethodBase request = null; // Objeto para realizar las peticiones HTTP GET o POST
21.         int status = 0; // Código de la respuesta HTTP
22.         BufferedReader reader = null; // Se usa para leer la respuesta a la petición
23.         String line = null; // Se usa para leer cada una de las líneas de texto d
24.
25.         try {
26.             // Instanciamos el objeto
27.             httpClient = new HttpClient();
28.
29.             // Invocamos por POST
30.             request = new PostMethod("http://127.0.0.1:8080/axis2/services/Version/getVersion" );
31.
32.             // Añadimos Los parámetros que deseemos a la petición
33.             ((PostMethod) request).addParameter( "parametro1", "valor1");
34.             ((PostMethod) request).addParameter( "parametro2", "valor2");
35.
36.             // Indicamos reintente 2 veces en caso de que haya errores.
37.             request.getParams().setParameter(HttpMethodParams.RETRY_HANDLER, new DefaultHttpMethodRetryHand
38.
39.             // Añadimos Las cabeceras personalizadas que se requieran, de la siguiente forma.
40.             request.setRequestHeader( "HeadName", "HeadValue");
41.
42.             // Leemos el código de la respuesta HTTP que nos devuelve el servidor
43.             status = httpClient.executeMethod(request);
44.
45.             // Vemos si La petición se ha realizado satisfactoriamente
46.             if (status != HttpStatus.SC_OK) {
47.                 System.err.println( "Error\t" + request.getStatusCode() + "\t" + request.getStatusText() +
48.             } else {
49.                 // Leemos el contenido de la respuesta y realizamos el tratamiento de la mism
50.                 // En nuestro caso, simplemente mostramos el resultado por la salida estándar
51.                 reader = new BufferedReader(new InputStreamReader(request.getResponseAsStream(), request
52.                 line = reader.readLine();
53.                 while (line != null) {
54.                     System.out.println(line);
55.                     line = reader.readLine();
56.                 }
57.             }
58.             catch (Exception ex){
59.                 System.err.println( "Error\t: " + ex.getMessage());
60.                 ex.printStackTrace();
61.             } finally {
62.                 // Liberamos La conexión. (También libera Los stream asociados)
63.                 request.releaseConnection();
64.             }
65.         }
66.     }
67.

```

Cuando Axis recibe una petición REST, internamente la traduce, a SOAP, por ejemplo:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" >
  <soapenv:Body>
    <axis2:getVersion xmlns:axis2="http://ws.apache.org/goGetWithREST" />
  </soapenv:Body>
</soapenv:Envelope>

```

Información que nos devuelve el Servicio Web en todos los caso

```

view plain print ?
01. <ns:getVersionResponse xmlns:ns="http://axisversion.sample">
02.   <ns:return>Hello I am Axis2 version service , My version is 1.3 </ns:return>
03. </ns:getVersionResponse>

```

Conclusiones

Como podemos deducir de este tutorial, para hacer que un servicio Web sea RESTful es más un tema del motor de servicio Web (en este caso Axis) que del propio servicio.

Además también se puede deducir que el acceso a este tipo de servicios Web no tiene dependencias con contratos WSDL, clases específicas de comunicación, esquemas XML, etc. por lo que se hace más fácil y ligero su uso.

Un saludo, Carlos García.

- 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 subcripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Schemas XML. Introducción esquemas XML	Los esquemas XML (schemas XML) son una evolución natural de las DTDs. Os mostramos como emprezar con esta tecnología.	2003-12-17	17625	pdf
XML y XSL en Cliente	En este tutorial os enseñamos como formaterar documentos XML directamente en vuestro navegador a través de Plantillas XSL. En cursos sucesivos veremos como hacerlo en el servidor, para no crear dependencias con el navegador del cliente.	2003-06-11	16069	pdf
Transformación de XML y XSL en JSPs	Os mostramos como poder utilizar XML y XSL en JSPs, combinado con el Patrón MVC	2003-12-06	25683	pdf
WebServices con Axis y JBoss	En este tutorial os mostramos como realizar servicios web utilizando Axis y el contenedor de aplicaciones web JBoss	2006-04-03	16068	pdf
XML Encryption, Criptografía sobre XML	Carlos García describe en este tutorial la encriptación de secciones de documentos XML utilizando el lenguaje del W3C: XML Encryption	2008-04-03	177	pdf
Trabajando con Axis	Utilizando Apache Axis, os mostramos otro interesante tutorial que ilustra su utilización para implementar web services	2006-04-07	27426	pdf
Uso básico de AmberPoint express con Axis	En este tutorial se muestra cómo utilizar AmberPoint Express para monitorizar un servicio web que hayamos hecho nosotros, y se cuentan algunas cosas inesperadas que he encontrado.	2006-11-02	3326	pdf
XML básico	Si quieres ver de un modo visual como crear un documento XML, este es tu tutorial. Este es el primero de un conjunto de tutoriales que iremos publicando sobre esta fascinante y amplia tecnología	2003-06-10	21702	pdf
XML Signature - Firma Digital sobre XML	En este tutorial aprenderemos a firmar digitalmente y validar un documento utilizando la implementación de Apache.	2008-04-03	175	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.