

# ¿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)

- **@GET**: Indica que el método anotado corresponde a una petición HTTP GET.
- **@POST**: Indica que el método anotado corresponde a una petición HTTP POST.
- **@HeaderParam**: Enlaza una cabecera http al parámetro de un método.
- **@HttpMethod**: Asocia un método con el nombre de un método HTTP .
- **@Path**: Identifica la URI de una clase o método que sirve las peticiones.
- **@ProduceMime**: Define el/los tipo(s) MIME que los métodos producen.
- **@QueryParam**: Enlaza un parámetro de la petición HTTP con un parámetro del método java anotado.
- ... más sobre el api en <https://jsr311.dev.java.net/nonav/releases/0.6/index.html>

### Veamos un ejemplo

Usaremos el ejemplo del webservice que sumaba números de los tutoriales anteriores, al que añadiremos las anotaciones pertinentes.

¡Ojo!. No es necesario usar RESTFull con una clase que implementa un webservice. Podría ser otra clase, pero ilustraremos el ejemplo con una clase que también es webservice porque así os puedo comentar algunas problemillas que he encontrado en este caso:

```

...
@WebService
@Path("/numbers")
public class AddNumbersImpl {

    @GET
    @Path("/add")
    @ProduceMime("text/html")
    @WebMethod
    public int addNumbers(@QueryParam("n1") int number1, @QueryParam("n2") int number2) throws AddNumbersException {

        if (number1 < 0 || number2 < 0) {
            throw new AddNumbersException("Negative number cant be added! Numbers: " + number1 + ", " + number2);
        }
        return number1 + number2;
    }
}
...

```

**OTROS SERVICIOS 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.

Están remarcas las anotaciones nuevas para RESTFull:

**@Path** va a indicar que para acceder a los métodos de la clase por rest la petición debe hacerse a <http://.../numbers/>...

**@GET** indica que `addNumbers` sólo puede ser llamado por HTTP GET (si no especificamos ningún método HTTP `addNumbers` será inaccesible mediante REST).

**@Path** en el método indica que para acceder al recurso por REST debemos dirigir la petición a <http://.../numbers/add>.

**@ProduceMime** indica que el resultado de `addNumbers` es texto html (para que el navegador o el cliente http sepa interpretar la respuesta).

**@QueryParam** indica que el parámetro `number1` se debe mandar como `â€œn1=7&...` en la petición HTTP: <http://.../numbers/add?n1=7&...>

Bastante facilito, ¿no creéis?.

### Consideraciones adicionales

Puede ocurrir que según el tipo de dato que devuelva el método REST no sepa cómo interpretarlo y devolverlo al cliente (por ejemplo un método que genera cierto texto y queremos devolverlo como un pdf).

O al contrario, se envía un parámetro a un método que REST no sabra cómo tratar (por ejemplo una imagen).

Debemos entonces generar clases que sepan como escribir y cómo leer tipos de datos de las respuestas / peticiones HTTP.

### Si anotamos una clase como webservice y REST existen problemas para tipos de datos de retorno distintos de String.

En este caso debemos crear clases que implementen:

- `javax.ws.rs.ext.MessageBodyWriter<T>` ó
- `javax.ws.rs.ext.MessageBodyReader<T>`

y anotarlas con `@Provider`.

Opcionalmente se puede anotar la clase con `@ProduceMime` para indicar el tipo MIME para el que se puede usar la clase.

De esta forma generamos clases convertidoras de tipos de datos a un tipo MIME (veamos un ejemplo que convierte de la clase Number a html):

```
@Provider
@ProduceMime("text/html")
public class NumericMessageBodyWriter implements MessageBodyWriter<Number> {

    /** devuelve el tamaño en bytes de la respuesta */
    public long getSize(Number t) {
        return t.toString().getBytes().length;
    }

    /** devuelve si el tipo de dato que llega se puede escribir con
    esta clase */
    public boolean isWriteable(Class<?> type) {
        return Number.class.isAssignableFrom(type);
    }

    /** escribe el dato en la respuesta */
    public void writeTo(Number t, MediaType mediaType,
        MultivaluedMap<String, Object> httpHeaders,
        OutputStream entityStream) throws IOException {

        entityStream.write(("<h2>" + t.toString() + "</h2>").getBytes());
        entityStream.flush();
    }
}
```

## Enlaces de interés

METRO: <http://metro.dev.java.net>

JAX-WS: <http://jax-ws.dev.java.net>

WSIT: <http://wsit.dev.java.net>

REST: <http://jersey.dev.java.net>

Glassfish: <http://glassfish.dev.java.net>

<http://blogs.sun.com/theaquarium/>

## Conclusión

Espero que esta serie de 3 tutoriales os haya servido para ver otras opciones de generar webservice (y no sólo con axis 2).

Como ya os he dicho anteriormente, el uso de las anotaciones en este tipo de tareas me parece increíble, facilitando las tareas que anteriormente eran más complejas.

Ya sabéis que si tenéis algún problema, ya sea con Metro, Jersey o cualquier otra tecnología, no dudeis en contactar con nosotros, <http://www.autentia.com>.

- 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. 



SOME RIGHTS RESERVED 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](mailto:info@autentia.com)

## Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de suscripción a novedades:

E-mail

### Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	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	3323	pdf
Servicios Web RESTful en Axis 2	En este tutorial vamos realizar una descripción de REST y vamos a ver un ejemplo práctico de un Servicio Web RESTful en AXIS 2	2008-04-03	115	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	16051	pdf
Creación e invocación de Webservices por SSL	En este tutorial se pretende enseñar a desplegar un webservice usando SSL y a invocarlo correctamente	2006-11-29	8081	pdf
Generador automático de Webservices	Os mostramos como crear un servicio Web a partir de una clases, gracias a generadores automáticos de código y NetBeans	2003-10-16	33801	pdf
Trabajando con Axis	Utilizando Apache Axis, os mostramos otro interesante tutorial que ilustra su utilización para implementar web services	2006-04-07	27390	pdf
Axis2. Ejemplo de creación de un servicio Web	En este tutorial veremos como crear un servicio web a partir de un interface Java así como otros aspectos de está tecnología.	2008-04-04	84	pdf
Metro: pila de webservices de Sun. Integración con Maven 2	En este tutorial Germán nos enseñara a integrar la generación de webservices con Metro y Maven2.	2008-04-05	5	pdf
Metro: pila de webservices de Sun.	NE este tutorial Germán nos enseñara qué es y cómo usar Metro: pila de webservices de Sun en nuestras aplicaciones	2008-04-05	7	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](mailto:rcanales@adictosaltrabajo.com) para su resolución.