

¿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)

adictos al trabajo

Encuentra un trabajo que te guste y no volverás a trabajar ni un sólo día de tu vida
Confucio

Autentia
business solutions

patrocinado por **EDADOS**

E-mail:

Contraseña:

[Deseo registrarme](#)
He olvidado mis datos de acceso

- [Inicio](#)
- [Quiénes somos](#)
- [Tutoriales](#)
- [Formación](#)
- [Comparador de salarios](#)
- [Nuestro libro](#)
- [Charlas](#)
- [Más](#)

Estás en: [Inicio](#) [Tutoriales](#) [Trabajando con los Web Services de Liferay](#)



DESARROLLADO POR:
[Rubén Aguilera Díaz-Heredero](#)

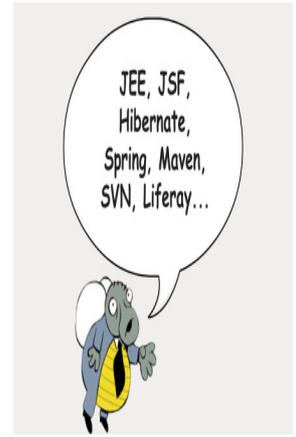
Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática, especialidad en Ingeniería del Software

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

Catálogo de servicios
[Autentia](#)



- [Anuncios Google](#)
- [Java](#)
- [Liferay Portal Hosting](#)
- [Java КНИГИ Jar](#)

Fecha de publicación del tutorial: 2009-02-26 16

Share | [Regístrate para votar](#)

Trabajando con los Web Services de Liferay

0. Índice de contenidos.

- 1. Entorno
- 2. Vamos al lío
- 3. Creación de un cliente de prueba
- 4. Conclusiones

1. Entorno

Este tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Mac Book Pro 17" (2,6 Ghz Intel Core i7, 8 GB DDR3)
- Sistema Operativo: Mac OS X Snow Leopard 10.6.4
- Liferay 6.0.5

2. Vamos al lío

Liferay publica toda su lógica de negocio a través de web services, podemos ver la lista de servicios web, una vez arrancado el servidor, en la URL: `http://localhost:8080/tunnel-web/axis`, previamente hemos tenido que añadir estas líneas a nuestro fichero `portal-ext.properties` con el fin de habilitar el acceso a estos servicios de una forma controlada.

[view plain](#) [copy to clipboard](#) [print](#) ?

```
01. axis.servlet.hosts.allowed=127.0.0.1,SERVER_IP,0:0:0:0:0:0:1%0
02. axis.servlet.https.required=false
```

Últimas Noticias

- [Java Thread Java Thread](#)

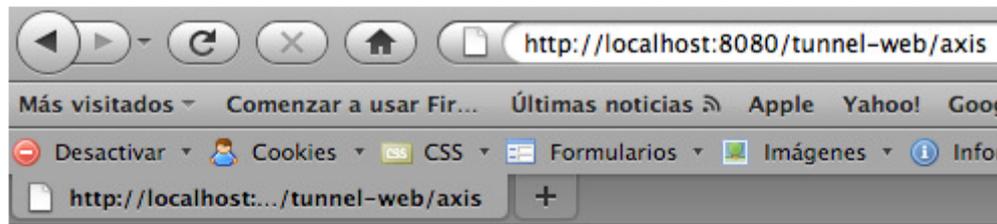
- [Reflexiones sobre AOS2010](#)
- [Comentando el libro: Nunca comas solo de Keith Ferrazzi y Tahl Raz.](#)
- [XII Charla Autentia - LiquiBase - iiiEmisión en Directo!!!](#)
- [Actualización en los esquemas del tutorial: "Cómo alcanzar el éxito en el sector de la informática"](#)
- [Comentado: Ingeniería de Software Ágil de E.M. Jimenez](#)

[Histórico de NOTICIAS](#)

Últimos Tutoriales

[Parte 1. Aprendiendo HTML](#)

En esta página podemos ver todos los servicios que tenemos disponibles junto su definición wsdl para poder crear nuestros propios clientes.



And now... Some Services

- Portal_AccountService ([wsdl](#))
- Portlet_IG_IGImageService ([wsdl](#))
 - deleteImage
 - deleteImageByFolderIdAndNameWithExtension
 - getGroupImages
 - getGroupImagesCount
 - getImageByFolderIdAndNameWithExtension
 - getImageByLargeImageId
 - getImageBySmallImageId
 - getImages
 - getImages
 - getImagesCount
 - getImage
- Portlet_SC_SCFrameworkVersionService ([wsdl](#))
 - addFrameworkVersion
 - deleteFrameworkVersion
 - getFrameworkVersion
 - getFrameworkVersions
 - getFrameworkVersions
 - updateFrameworkVersion
- Portlet_Shopping_ShoppingItemService ([wsdl](#))
 - addBookItems
 - deleteItem

Pinchando por ejemplo en el wsdl del primer servicio podríamos ver algo similar a esto:

para crear una página web

Cómo retocar una foto con Adobe Photoshop

Java i18n Properties Encoding

Construcción de componentes en wuija por composición

Reunión Madrid Ágil 02-11-2010: DDD (Domain Driven Design)

Últimos Tutoriales del Autor

Liferay IDE

CAS: Validador personalizado

CAS: Personalización de la interfaz

Introducción a CAS

Prey, localizador de dispositivos móviles robados

Síguenos a través de:



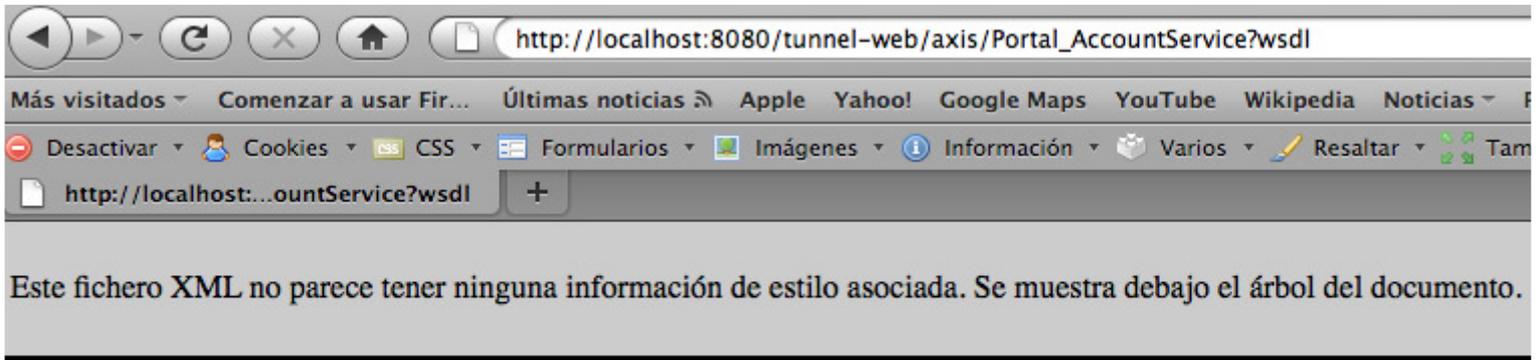
Últimas ofertas de empleo

2010-10-11
 Comercial - Ventas - SEVILLA.

2010-08-30
 Otras - Electricidad - BARCELONA.

2010-08-24
 Otras Sin catalogar - LUGO.

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



```

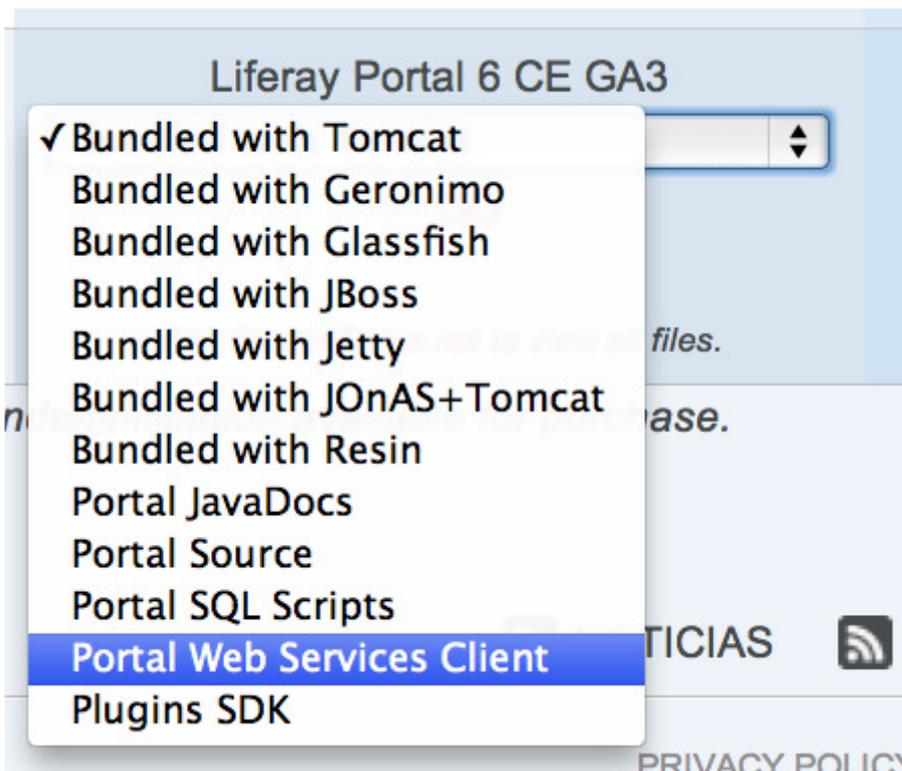
- <wsdl:definitions targetNamespace="urn:http.service.portal.liferay.com">
  - <!--
    WSDL created by Apache Axis version: 1.4Built on Apr 22, 2006 (06:55:48 PDT)
  -->
  <wsdl:portType name="AccountServiceSoap"/>
  - <wsdl:binding name="Portal_AccountServiceSoapBinding" type="impl:AccountServiceSoap">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  </wsdl:binding>
  - <wsdl:service name="AccountServiceSoapService">
    - <wsdl:port binding="impl:Portal_AccountServiceSoapBinding" name="Portal_AccountService">
      <wsdlsoap:address location="http://localhost:8080/tunnel-web/axis/Portal_AccountService"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

Podríamos utilizar este descriptor como base para la creación de clientes que consumieran este servicio. Para ello, Eclipse o Netbeans ofrecen wizards que generan todo el código necesario para realizar la llamada y formatear la respuesta de estos servicios.

Aunque Liferay ya hace esto por nosotros y nos lo empaqueta en un fichero llamado portal-client.jar que contiene las clases generadas y preparadas para acceder a todos los servicios que se mostraron.

Para la última versión de Liferay podemos descargar este .jar de su propia página de descargas en la dirección <http://www.liferay.com/es/downloads> con el nombre "Portal Web Services Client", que tiene el portal-client.jar más todas las dependencias necesarias.



Es muy importante que descargemos la versión que coincida con la versión del portal que estemos utilizando, ya que de una a otra puede variar significativamente. Para el resto de versiones tenemos

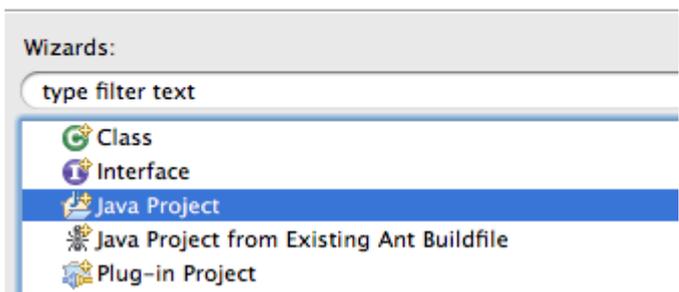
que recurrir a la página de sourceforge <http://sourceforge.net/projects/lportal/files/>

3. Creación de un cliente de prueba

Ahora vamos a crear un cliente en Java que consuma uno de estos servicios. Para ello, abrimos Eclipse y creamos un proyecto de Java para escritorio.

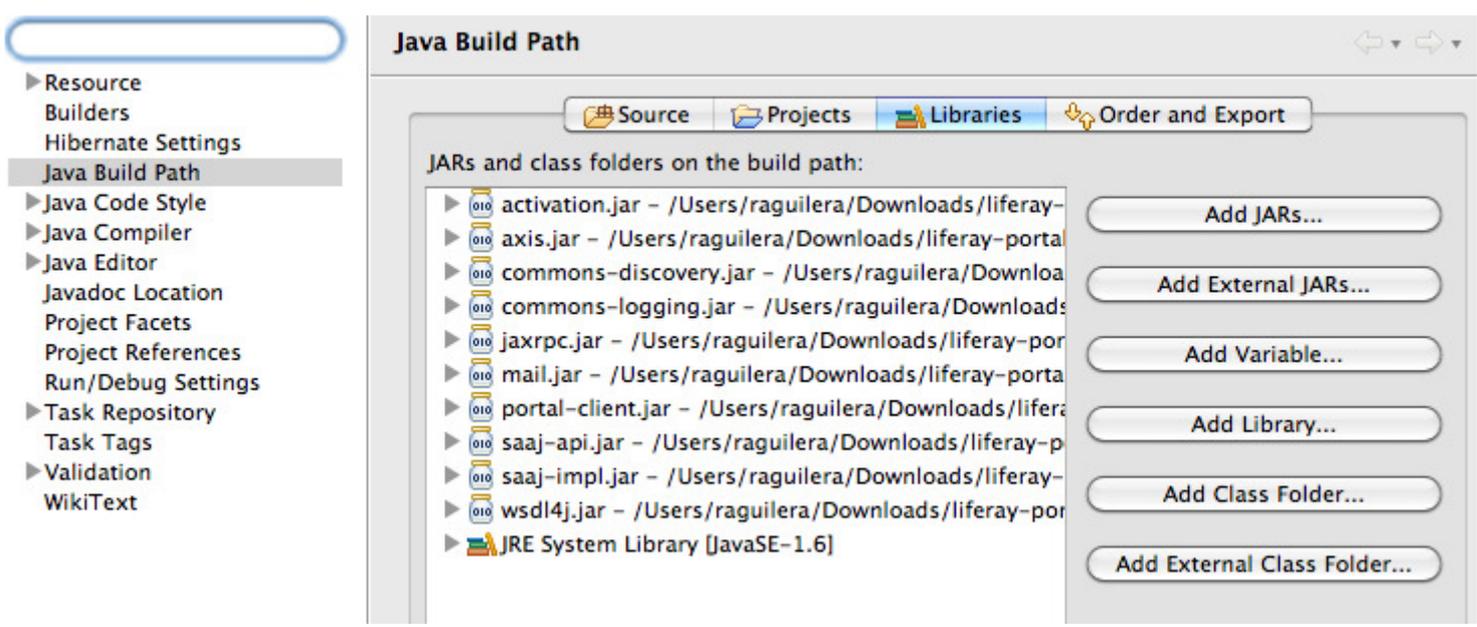
Select a wizard

Create a Java project



Pulsando en "Next" se muestra otra pantalla donde damos un nombre a nuestro proyecto por ejemplo prueba-cliente-ws y pulsamos en "Finish".

Ahora vamos a añadir nuestras dependencias al proyecto. Para ello con botón derecho sobre la raíz del proyecto seleccionamos "Build Path" --> "Configure Build Path" y en la pantalla que se muestra seleccionamos la pestaña "Libraries", pulsando en la opción "Add External JARs..." podemos seleccionar todos los .jar que hemos descargado anteriormente, con lo que la pantalla queda de esta forma:



Pulsando en "Ok" ya estamos en condiciones de implementar nuestro cliente que consuma un servicio web publicado por Liferay.

Entonces, con el botón derecho sobre la raíz del proyecto seleccionamos "New" -- "Class" e introducimos la siguiente información:

Java Class

Create a new Java class.

Source folder:

Package:

Enclosing type:

Name:

Modifiers: public default private protected
 abstract final static

Superclass:

Interfaces:

Which method stubs would you like to create?

public static void main(String[] args)
 Constructors from superclass
 Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

Generate comments

Pulsamos en "Finish" para comenzar a implementar el cliente. Para este ejemplo, vamos a utilizar el servicio web de recuperar un JournalArticle con el fin de mostrar la información por consola.

```

view plain copy to clipboard print ?
01. package com.autentia.ws.liferay;
02.
03. import java.net.MalformedURLException;
04. import java.net.URL;
05. import java.rmi.RemoteException;
06. import javax.xml.rpc.ServiceException;
07. import com.liferay.client.soap.portlet.journal.model.JournalArticleSoap;
08. import com.liferay.client.soap.portlet.journal.service.http.JournalArticleService;
09. import com.liferay.client.soap.portlet.journal.service.http.JournalArticleService;
10. import com.liferay.client.soap.portlet.journal.service.http.JournalArticleService;
11. import com.liferay.client.soap.portlet.journal.service.http.Portlet_Journal_Article_Soap;
12.
13. public class PruebaCliente {
14.     private static final String ID_JOURNAL_ARTICLE = "10455";
15.     private static final int GROUP_ID_JOURNAL_ARTICLE = 10156;
16.     private static final String PASSWORD_USUARIO_CONEXION_WS = "bruno";
17.     private static final String USUARIO_CONEXION_WS = "bruno@7cogs.com";
18.     private static final String URL_JOURNAL_ARTICLE_WS =
19.         "http://localhost:8080/tunnel-
20.         web/secure/axis/Portlet_Journal_JournalArticleService";
21.
22.     public static void main(String[] args) {
23.         // Establecemos la conexión con el servicio web
24.         JournalArticleServiceSoapService
25.         service = new JournalArticleServiceSoapServiceLocator();

```

```

25.     JournalArticleServiceSoap journalArticleService;
26.     try {
27.         journalArticleService = service.getPortlet_Journal_JournalArticle!
28.
29.         ((new URL(URL_JOURNAL_ARTICLE_WS)));
30.         ((Portlet_Journal_JournalArticleServiceSoapBindingStub)
31.         journalArticleService) .setUsername(USUARIO_CONEXION_WS);
32.         ((Portlet_Journal_JournalArticleServiceSoapBindingStub)
33.         journalArticleService) .setPassword
34.         (PASSWORD_USUARIO_CONEXION_WS);
35.
36.         //Utilizamos el método remoto getArticle para recuperar un
37.
38.         JournalArticleSoap journalArticleSoap = journalArticleService
39.         .getArticle(GROUP_ID_JOURNAL_ARTICLE, ID_JOURNAL_ARTICLE);
40.
41.         //Pintamos por consola la información
42.         System.out.println("Este es su título: " +
43.         journalArticleSoap.getTitle());
44.         System.out.println("Este es su tipo: "
45.         + journalArticleSoap.getType());
46.         System.out.println("Este es el id del
47.         autor: " + journalArticleSoap.getUserId());
48.         System.out.println("Esta es la fecha de modificación: " +
49.         journalArticleSoap.getModifiedDate().getTime());
50.     } catch (MalformedURLException e) {
51.         e.printStackTrace();
52.     } catch (ServiceException e) {
53.         e.printStackTrace();
54.     } catch (RemoteException e){
55.         e.printStackTrace();
56.     }
57. }

```

Al ejecutar la clase con botón derecho sobre el fichero "Run As..." -- "Java Application" obtenemos la siguiente salida por consola.

```

<terminated> PruebaCliente (1) [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.j
Este es su título: Welcome
Este es su tipo: general
Este es el id del autor: 10134
Esta es la fecha de modificación: Thu Nov 18 10:47:56 CET 2010

```

4. Conclusiones

Como era de esperar en un producto tan maduro como Liferay no es muy complicado interactuar con sus servicios desde clientes externos gracias a la utilización de web services. De esta forma podríamos, por ejemplo, crear un cliente en .NET que añadiera contenido web o creará usuarios desde otra aplicación totalmente independiente a Liferay.

Saludos.

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» **Registrate** y accede a esta y otras ventajas «

COMENTARIOS



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

Copyright 2003-2010 © All Rights Reserved | [Texto Completo](#) | [condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) |

