

¿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)

Estas en: [Inicio](#) [Tutoriales](#) Intercomunicación entre portlets.

Ultimas Noticias

- » VIII Charla Autentia - Alfresco Community Edition
- » Plasma Cars - Las fotos
- » VII Charla Autentia: Pluto - Vídeos y Material
- » Nueva sección - Fotos con el libro
- » Estuvimos en el evento de Liferay en Madrid
- » VII Charla Autentia - Pluto
- » Competición Plasma Cars (Autos Locos) - SEGUNDO INTENTO
- » Probando con Marick - Fotos y vídeo
- » Competición Plasma Cars (Autos Locos) - EVENTO

+Noticias Destacadas

- » VIII Charla Autentia - Alfresco Community Edition
- » Plasma Cars - Las fotos
- » VII Charla Autentia: Pluto - Vídeos y Material
- » Nueva sección - Fotos con el libro

+Comentarios Cómic

+Enlaces

Catálogo de servicios Autentia



Triptico
(6,3 MB)



Cómic
(3,1 MB)

Tutorial desarrollado por



Ángel García Jerez

Consultor tecnológico de desarrollo de proyectos informáticos. Co-autor del libro "Actualización y mantenimiento del PC (Edición de 2008)" publicado por Anaya Multimedia

Ingeniero Técnico en Informática de Sistemas e Ingeniero en Informática (premio al mejor expediente de su promoción)

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

[Descargar \(6,2 MB\)](#)

[Descargar en versión comic \(3,1 MB\)](#)

[AdictosAlTrabajo.com](#) es el Web de difusión de conocimiento de [Autentia](#).



[Catálogo de cursos](#)

Descargar este documento en formato PDF: [intercomunicacion.pdf](#)

Fecha de creación del tutorial: 2010-05-17

Share |

Intercomunicación entre portlets.

1. Introducción
2. Entorno
3. Ejemplo práctico
 - 3.1. Intercomunicación entre Portlets utilizando la sesión.
 - 3.2. Intercomunicación entre Portlets utilizando parámetros públicos.
 - 3.3. Intercomunicación entre Portlets utilizando eventos.
4. Conclusión

1. Introducción

La intercomunicación entre portlets es una funcionalidad que permite compartir o enviar información entre varios portlets desplegados en cualquier lugar del portal. Antiguamente con la versión 1 de la especificación JSR-168 la comunicación únicamente se realizaba a través de la Session del Portlet o utilizando la implementación que ofrecía el Gestor de Portales que se estuviera manejando. El uso de la solución propietaria impedía que dicho portlet pudiera ser exportado al resto de gestores de portales del mercado. Con la aparición de la versión 2 de la especificación JSR 286 se resolvió este problema. En esta versión se han definido dos nuevos métodos para intercambiar información entre los portlets desplegados en un portal: por eventos o por parámetros públicos de renderización. Con este tutorial se mostrará el uso de las tres formas a través de un pequeño ejemplo. Se utilizará Liferay como gestor de portales y se tomará como punto de partida el tutorial "[Utilización de arquetipos y plugins de LifeRay para Maven.](#)" donde se explican los pasos para crear un portlet en LifeRay.

2. Entorno

El tutorial está escrito usando el siguiente entorno:

- **Hardware:** Mac Book Pro (Core 2 Duo 2,8 Ghz, 4 GB RAM, 500 GB)
- **Sistema Operativo:** Snow Leopard
- **Navegador:** Mozilla Firefox 3.6.3
- **Maven:** 2.2.1
- **Liferay:** 5.2.3 + SDK.

3. Ejemplo práctico:

Para ver el funcionamiento de cada método de comunicación se realizará un ejemplo muy sencillo. Se crearán dos proyectos maven, el primero llamado "localizador-portlet" que se encargará de obtener una dirección a través de un formulario, y el segundo llamado "visualizador-portlet" mostrará un mapa con la localización de esa dirección. En cada uno de ellos se harán tres versiones del mismo portlet, implementando cada tipo de comunicación.

Lo primero de todo es crear los dos proyectos utilizando el arquetipo "liferay-portlet-archetype" tal como se indica en el tutorial al que se ha hecho referencia anteriormente.

Acceso de usuarios registrados:

E-mail:

Contraseña:

[Deseo registrarme](#)

[He olvidado mis datos de acceso](#)

Registra tu empresa:

Descubre las ventajas de registrar tu empresa en [AdictosAlTrabajo...](#)

[Registrar mi empresa](#)

[Listado de empresas ya registradas](#)



☐ Web

☒ [www.adictosaltrabajo.com](#)

Últimos tutoriales

2010-05-17
[Intercomunicación entre portlets.](#)

2010-05-17
[Actualización de bugzilla a la versión 3.6.](#)

2010-05-11
[eCobertura y EciEMMA: Plugins para la medición de cobertura del código.](#)

2010-05-11
[VirtualBox. Configuración de la conexión de red.](#)

```
mvn -e archetype:create -DgroupId=com.adictos.portlet -DartifactId=localizador-portlet -DarchetypeGroupId=com.liferay.archetype:create -DgroupId=com.adictos.portlet -DartifactId=visualizador-portlet -DarchetypeGroupId=com.liferay
```

3.1 Intercomunicación entre Portlets utilizando la sesión.

La intercomunicación utilizando la sesión fue el primer método que se utilizó para este propósito. Su uso es bastante sencillo, se recupera la sesión del portlet y se establece el atributo que se desea compartir, indicando que dicho atributo va a ser visible por el resto de la aplicación (APPLICATION_SCOPE).

Empezamos con el portlet "localizador-portlet". Su función es presentar un formulario para que el usuario introduzca una dirección. Cuando éste lo envíe, la dirección se almacenará en la sesión para que el portlet "visualizador-portlet" lo recupere y pinte el mapa con la localización.

Creamos una clase llamada LocalizadorSessionPortlet que extienda de GenericPortlet.

```
package com.adictos.portlet.localizador;

import java.io.IOException;

import javax.portlet.ActionRequest;
import javax.portlet.ActionResponse;
import javax.portlet.GenericPortlet;
import javax.portlet.PortletException;
import javax.portlet.PortletRequestDispatcher;
import javax.portlet.PortletSession;
import javax.portlet.ProcessAction;
import javax.portlet.RenderRequest;
import javax.portlet.RenderResponse;

public class LocalizadorSessionPortlet extends GenericPortlet{

    protected String viewJSP;
    protected String keyAddressSession;
    public static final String KEY_JSP_ADDRESS = "address";

    @Override
    public void init() throws PortletException {
        //Se recupera la jsp encargada de pintar el portlet
        viewJSP = getInitParameter("view-jsp");
        //Se recupera la clave utilizada para compartir la dirección entre los portlets
        keyAddressSession = getInitParameter("key-address-session");
    }

    //Método que se ejecuta cuando se envía el formulario
    @ProcessAction(name="localizar")
    public void processLocalizador(ActionRequest request, ActionResponse response) {
        //Se recupera la sesión del portlet
        PortletSession session = request.getPortletSession();
        //Se añade a la sesión la dirección en el ámbito de la aplicación
        session.setAttribute(keyAddressSession, request.getParameter(getDefaultNamespace()+ KEY_JSP_ADDRESS),PortletSession.APPLICATION_SCOPE);
    }

    @Override
    public void doView(RenderRequest request, RenderResponse response)throws IOException, PortletException {
        include(viewJSP, request, response);
    }

    protected void include(
        String path, RenderRequest renderRequest,
        RenderResponse renderResponse)
        throws IOException, PortletException {

        PortletRequestDispatcher portletRequestDispatcher =
            getPortletContext().getRequestDispatcher(path);

        if (portletRequestDispatcher == null) {
            throw new IOException(path + " is not a valid include");
        }
        else {
            portletRequestDispatcher.include(renderRequest, renderResponse);
        }
    }
}
```

Creamos la jsp view.jsp que será común a todos los portlets que se vayan a crear:

```
<%@ taglib uri="http://java.sun.com/portlet" prefix="portlet" %>
<%@page import="com.adictos.portlet.localizador.LocalizadorSessionPortlet"%>

<%@page import="javax.portlet.Portlet"%>
<%@page import="javax.portlet.ActionRequest"%><portlet:defineObjects />

<portlet:actionURL name="localizar" var="actionForm"/>

<form action="<%=actionForm%>" method="post">
    <label for="<portlet:namespace/><%=LocalizadorSessionPortlet.KEY_JSP_ADDRESS%>">Dirección:</label>
    <input type="text" name="<portlet:namespace/><%=LocalizadorSessionPortlet.KEY_JSP_ADDRESS%>" />
    <input type="submit" />
</form>
```

Se añade la definición del portlet al descriptor portlet.xml:

2010-05-11
Múltiples datasources en JasperReports

2010-05-10
Utilización de arquetipos y plugins de LifeRay para Maven.

2010-05-07
Instalar un componente en Talend Open Studio (TOS)

2010-05-06
Talend Open Studio (TOS) 4.0

2010-05-05
Gestión de los Requisitos (y II) : los casos de uso

2010-05-05
Gestión de los requisitos

2010-05-04
Declaración de IVA trimestral en la AEAT por Internet

2010-05-04
Certificados en Firefox (FNMT y AEAT)

2010-04-26
JCaptcha - Generación de Captchas en Java

2010-04-23
Instalar Puesto PHP-Java en Tomcat

2010-04-22
AppWidget Android: Ejemplo usando BroadcastReceiver y Localización

2010-04-20
Facelets en JSF 2: sistema de plantillas y componentes por composición.

2010-04-19
DbVisualizer free version.

2010-04-09
Session Timeout en RichFaces, con el soporte de Jboss Seam.

2010-04-08
Jetspeed-2 de Apache Software Foundation

2010-04-07
Primeros pasos con Balsamiq Mockups

2010-03-18
Revisando los ejemplos de Cocos2d para iPhone.

2010-03-16
Organización de eventos con StageHQ

```
<portlet-app
  version="2.0"
  xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd http://java.sun.com/xml/ns/portle
>
...
  <portlet>
    <portlet-name>localizador-portlet</portlet-name>
    <display-name>localizador-portlet</display-name>
    <portlet-class>com.adictos.portlet.localizador.LocalizadorSessionPortle</portlet-class>
    <init-param>
      <name>view-jsp</name>
      <value>/view.jsp</value>
    </init-param>
    <init-param>
      <name>key-address-session</name>
      <value>com.adictos.portlet.localizador.key.address</value>
    </init-param>
    <expiration-cache>0</expiration-cache>
    <supports>
      <mime-type>text/html</mime-type>
    </supports>
    <portlet-info>
      <title>localizador-portlet</title>
      <short-title>localizador-portlet</short-title>
      <keywords>localizador-portlet</keywords>
    </portlet-info>
    <security-role-ref>
      <role-name>administrator</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>guest</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>power-user</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>user</role-name>
    </security-role-ref>
  </portlet>
...
</portlet-app>
```

A continuación se añade la configuración específica del portlet para que funcione correctamente en liferay. En el descriptor liferay-portlet.xml se definen algunos parámetros específicos para el gestor de portales. Debemos hacer hincapié en el atributo private-session-attributes que hay que deshabilitar para que la compartición de atributos a través de la sesión funcione.

```
<liferay-portlet-app>
...
  <portlet>
    <portlet-name>localizador-portlet</portlet-name>
    <icon>/icon.png</icon>
    <instanceable>true</instanceable>
    <!-- se le indica a liferay que el portlet va a compartir los atributos de la sesion con otros portlets
    <private-session-attributes>>false</private-session-attributes>
    <header-portlet-css>/css/portlet.css</header-portlet-css>
    <footer-portlet-javascript>/js/javascript.js</footer-portlet-javascript>
  </portlet>
...
</liferay-portlet-app>
```

Y por último se añade al fichero liferay-display.xml la asociación del portlet con la categoría a la que pertenece. En nuestro caso a la categoría "Sample".

```
<display>
  <category name="category.sample">
    ...
    <portlet id="localizador-portlet" />
    ...
  </category>
</display>
```

Ahora se tiene que desarrollar el portlet "visualizador-portlet", su función es recoger la dirección que haya depositado en la sesión "localizador-portlet" y utilizando el API de Google Maps mostrar el mapa con la localización.

Creamos la clase VisualizadorSessionPortlet:

2010-03-15

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

2010-03-15

Optimización de páginas web con Page Speed.

2010-03-09

JSF 2 ya está aquí !!! The JSF Return, ahora más sencillo que nunca !!!

2010-03-08

Instalación de tus programas en tu iPhone.

2010-03-04

Sacar Release de un proyecto con Maven

2010-03-03

Instalación de Subversion y Apache en Ubuntu

2010-03-03

Cómo instalar la JDK de SUN en Fedora Linux

2010-03-02

Creando un botón de compra de Paypal con datos cifrados

2010-03-01

Creación de un plugin de tipo hook en Liferay

2010-03-01

ScrumCards de Autentia en Android

2010-02-25

Creando la baraja de SCRUM de Autentia como aplicación para Android

2010-02-25

Instalar CentOS en Virtualbox con NetInstall

2010-02-22

Expresiones CRON

2010-02-19

Cómo utilizar el DataStore de Google App Engine con JDO

2010-02-19

Recursos Freeware

2010-02-17

Plugin de mejora de graficos para JMeter

2010-02-17

Cómo utilizar el datastore de Google App Engine con su API de nivel inferior

2010-02-16

Aprendiendo Objective-C desarrollando para nuestro Iphone 3Gs

```
package com.adictos.portlet.visualizador;

import java.io.IOException;

import javax.portlet.GenericPortlet;
import javax.portlet.PortletException;
import javax.portlet.PortletRequestDispatcher;
import javax.portlet.PortletSession;
import javax.portlet.RenderRequest;
import javax.portlet.RenderResponse;

public class VisualizadorSessionPortlet extends GenericPortlet {

    protected String viewJSP;
    protected String keyAddressSession;
    public static final String KEJ_JSP_ADDRESS = "address";

    @Override
    public void init() throws PortletException {
        //Se recupera La jsp encargada de pintar el portlet
        viewJSP = getInitParameter("view-jsp");
        //Se recupera La clave utilizada para compartir la dirección entre los portlet
        keyAddressSession = getInitParameter("key-address-session");
    }

    @Override
    protected void doView(RenderRequest request, RenderResponse response)
        throws PortletException, IOException {
        //Se recupera La sesión del portlet
        PortletSession session = request.getPortletSession(false);
        //Se recupera La dirección que haya establecido el otro portlet en La sesión
        String address = (String) session.getAttribute(keyAddressSession, PortletSession.APPLICATION_SCOPE);
        if (address != null) {
            //Se establece La dirección en La request donde lo leerá la js
            request.setAttribute(KEJ_JSP_ADDRESS, address);
        }

        include(viewJSP, request, response);
    }

    @Override
    protected void doHeaders(RenderRequest request, RenderResponse response) {
        super.doHeaders(request, response);
    }

    protected void include(
        String path, RenderRequest renderRequest,
        RenderResponse renderResponse)
        throws IOException, PortletException {

        PortletRequestDispatcher portletRequestDispatcher =
            getPortletContext().getRequestDispatcher(path);

        if (portletRequestDispatcher == null) {
            throw new IOException(path + " is not a valid include");
        }
        else {
            portletRequestDispatcher.include(renderRequest, renderResponse);
        }
    }
}
```

Creamos la jsp view.jsp común a todas las versiones del portlet:

```
<%@ taglib uri="http://java.sun.com/portlet" prefix="portlet" %>
<%%page import="com.adictos.portlet.visualizador.VisualizadorSessionPortlet"%>

<portlet:defineObjects />

Dirección seleccionada: <strong><%=request.getAttribute(VisualizadorSessionPortlet.K EJ_JSP_ADDRESS)></strong>
<div id="address" style="visibility: hidden"><%=request.getAttribute(VisualizadorSessionPortlet.K EJ_JSP_ADDRESS)>
<div id="google_maps" style="width:100%; height: 400px;">
</div>
```

Creamos el fichero javascript.js que se usa para localizar la dirección en Google Maps (idéntico en todos los ejemplos).

2010-02-11
Introducción a JCL.

2010-02-09
Creando la Baraja de SCRUM de Autentia como aplicación para el iPhone 3G.

2010-02-08
Cómo generar versiones imprimibles de páginas web

2010-02-04
Como cambiar el tamaño de las fuentes en Xcode (el entorno de desarrollo para Mac e iPhone)

2010-02-04
Primeros pasos con Enterprise Architect y UML 2.x

2010-02-04
Creación de un componente JSF, basádonos en un plugin de jQuery, con el soporte de RichFaces.

2009-02-03
Sincronizando el Mail de Mac con Gmail, el correo de Google

2010-02-03
Integración de jQuery en RichFaces.

2010-02-02
AjaxSingle: el partialSubmit de RichFaces.

2010-02-01
Introducción a RichFaces.

Últimas ofertas de empleo

2010-04-28
Comercial - Compras - CORDOBA.

2010-04-25
Otras Sin catalogar - MADRID.

2010-04-25
Atención a cliente - Call Center - MADRID.

2010-04-21
Comercial - Ventas - MADRID.

2009-06-25
Atención a cliente - Call Center - BARCELONA.

Anuncios Google

```
Event.observe(window,"load", initGMaps);
```

```
var map;

function onLoadGeo (point) {
    map.setCenter(point,16);
    map.clearOverlays() ;
    map.addOverlay(new GMarker(point));
}

function initGMaps() {
    if (GBrowserIsCompatible()) {
        map = new GMap2($('google_maps'));
        map.setCenter(new GLatLng(39.856778,-4.024476),4);
        map.setMapType(G_NORMAL_MAP);
        var geo = new GClientGeocoder();

        geo.getLatLng($('address').firstChild.nodeValue,onLoadGeo);

    }
}
```

Y por último se añade la configuración del portlet en los xml. Remarcar las etiquetas del fichero liferay-portlet.xml footer-portal-javascript y footer-portlet-javascript pues son el método que pone a disposición liferay para añadir recursos javascript con el fin de ser usados en el portlet.

Fichero portlet.xml

```
<portlet-app
  version="2.0"
  xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd http://java.sun.com/xml/ns/portlet-
>
...
  <portlet>
    <portlet-name>visualizador-portlet</portlet-name>
    <display-name>visualizador-portlet</display-name>
    <portlet-class>com.adictos.portlet.visualizador.VisualizadorSessionPortle</portlet-class>
    <init-param>
      <name>view-jsp</name>
      <value>/view.jsp</value>
    </init-param>
    <init-param>
      <name>key-address-session</name>
      <value>com.adictos.portlet.localizador.key.address</value>
    </init-param>
    <expiration-cache>0</expiration-cache>
    <supports>
      <mime-type>text/html</mime-type>
    </supports>
    <portlet-info>
      <title>visualizador-portlet</title>
      <short-title>visualizador-portlet</short-title>
      <keywords>visualizador-portlet</keywords>
    </portlet-info>
    <security-role-ref>
      <role-name>administrator</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>guest</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>power-user</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>user</role-name>
    </security-role-ref>
  </portlet>
...
</portlet-app>
```

Fichero liferay-portlet.xml:

```
<liferay-portlet-app>
...
  <portlet>
    <portlet-name>visualizador-portlet</portlet-name>
    <icon>/icon.png</icon>
    <instanceable>true</instanceable>
    <private-session-attributes>false</private-session-attributes>
    <header-portlet-css>/css/portlet.css</header-portlet-css>
    <!-- recursos javascript que utilizará el portlet -->
    <footer-portal-javascript>http://maps.google.com/maps?file=api&v=2&key=ABQIAAAAMEZHKeOqK-v_PHwvq2QJRTFY3:
    <footer-portlet-javascript>/js/prototype.js</footer-portlet-javascript>
    <footer-portlet-javascript>/js/javascript.js</footer-portlet-javascript>
  </portlet>
...
</liferay-portlet-app>
```

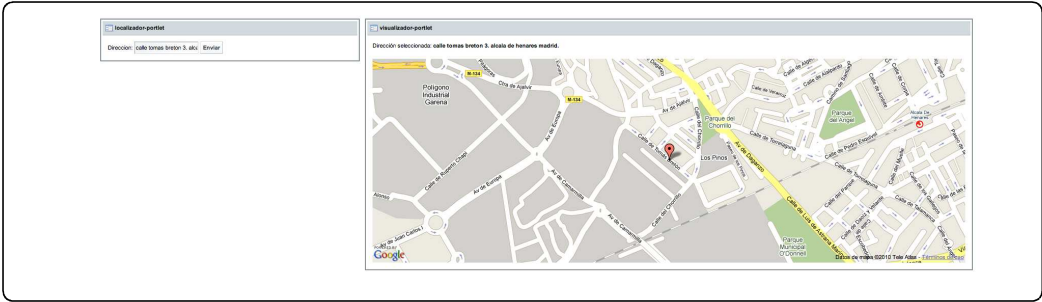
Fichero liferay-display.xml:


```
<display>
  <category name="category.sample">
    ...
    <portlet id="visualizador-portal" />
    ...
  </category>
</display>
```

Después se despliegan los portlets en liferay; para ello se ejecuta el comando en cada uno de los proyectos maven generados:

```
mvn -e clean package liferay:deploy
```

Finalmente se accede al portal como administrador, se añaden los portlets a una página y se realiza una prueba enviando el formulario con una dirección.



3.2 Intercomunicación entre Portlets utilizando parámetros públicos.

La intercomunicación utilizando parámetros públicos es un método disponible a partir de la versión 2 (JSR 286); para poder utilizarlo únicamente se debe registrar el parámetro en el fichero portlet.xml con el atributo "supported-public-render-parameter". En ese momento, cuando en la url de invocación se encuentre dicho parámetro, aquellos portlets que hayan indicado su uso podrán acceder a su valor de la misma forma que se hace con cualquier otro parámetro.

En este ejemplo y en el siguiente sólo se van a incluir aquellos recursos relevantes para el mismo, ya que la información de los ficheros liferay-display.xml, liferay-portal.xml es prácticamente idéntica a excepción del nombre del portlet.

Para comenzar crea la clase LocalizadorParameterPortlet:

```

package com.adictos.portlet.localizador;

import java.io.IOException;

import javax.portlet.ActionRequest;
import javax.portlet.ActionResponse;
import javax.portlet.GenericPortlet;
import javax.portlet.PortletException;
import javax.portlet.PortletRequestDispatcher;
import javax.portlet.ProcessAction;
import javax.portlet.RenderRequest;
import javax.portlet.RenderResponse;

public class LocalizadorParameterPortlet extends GenericPortlet {

    protected String viewJSP;
    protected String keyAddressSession;
    public static final String KEY_JSP_ADDRESS = "address";

    @Override
    public void init() throws PortletException {
        //Se recupera la jsp encargada de pintar el portlet
        viewJSP = getInitParameter("view-jsp");
    }
    //Se define el nombre del accion que ejecutara este metodo
    @ProcessAction(name="localizar")
    public void processLocalizador(ActionRequest request, ActionResponse response) {

        //Se recupera la direccion enviada desde el formulario
        String address = request.getParameter(getDefaultNamespace()+ KEY_JSP_ADDRESS);
        if (address != null) {
            //Establecemos el parametro render publico
            response.setRenderParameter("address", address);
        }

    }

    @Override
    public void doView(RenderRequest request, RenderResponse response) throws IOException, PortletException {
        include(viewJSP, request, response);
    }

    protected void include(
        String path, RenderRequest renderRequest,
        RenderResponse renderResponse)
        throws IOException, PortletException {

        PortletRequestDispatcher portletRequestDispatcher =
            getPortletContext().getRequestDispatcher(path);

        if (portletRequestDispatcher == null) {
            throw new IOException(path + " is not a valid include");
        }
        else {
            portletRequestDispatcher.include(renderRequest, renderResponse);
        }
    }
}

```

Se añade la definición del portlet al descriptor portlet.xml, donde se establece el parámetro público de renderización:


```

<portlet-app
  version="2.0"
  xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd http://java.sun.com/xml/ns/portle
>
...
  <portlet>
    <portlet-name>localizador-parameter-portlet</portlet-name>
    <display-name>localizador-parameter-portlet</display-name>
    <portlet-class>com.adictos.portlet.localizador.LocalizadorParameterPortle</portlet-class>
    <init-param>
      <name>view-jsp</name>
      <value>/view.jsp</value>
    </init-param>
    <expiration-cache>0</expiration-cache>
    <supports>
      <mime-type>text/html</mime-type>
    </supports>
    <portlet-info>
      <title>localizador-parameter-portlet</title>
      <short-title>localizador-parameter-portlet</short-title>
      <keywords>localizador-parameter-portlet</keywords>
    </portlet-info>
    <security-role-ref>
      <role-name>administrator</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>guest</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>power-user</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>user</role-name>
    </security-role-ref>
    <supported-public-render-parameter>address</supported-public-render-parameter>
  </portlet>
  ...
  <public-render-parameter>
    <identifier>address</identifier>
    <qname xmlns:x="http://www.adictosaltrabajo.com/public-render-parameters">x:address</qname>
  </public-render-parameter>
  ...
</portlet-app>

```

Ahora creamos la versión del visualizador para que soporte parámetro públicos:

```

package com.adictos.portlet.visualizador;

import java.io.IOException;

import javax.portlet.GenericPortlet;
import javax.portlet.PortletException;
import javax.portlet.PortletRequestDispatcher;
import javax.portlet.RenderRequest;
import javax.portlet.RenderResponse;

public class VisualizadorParameterPortlet extends GenericPortlet {

    protected String viewJSP;
    public static final String KEJ_JSP_ADDRESS = "address";
    @Override
    public void init() throws PortletException {
        //Se recupera la jsp encargada de pintar el portlet
        viewJSP = getInitParameter("view-jsp");
    }

    @Override
    protected void doView(RenderRequest request, RenderResponse response)
        throws PortletException, IOException {
        //Se recupera el valor del parametro publico de renderizacio
        String address = request.getParameter("address");
        if (address != null) {
            //Se establece la direccion en la request donde lo leera la js
            request.setAttribute(KEJ_JSP_ADDRESS, address);
        }

        include(viewJSP, request, response);
    }

    @Override
    protected void doHeaders(RenderRequest request, RenderResponse response) {
        super.doHeaders(request, response);
    }

    protected void include(
        String path, RenderRequest renderRequest,
        RenderResponse renderResponse)
        throws IOException, PortletException {

        PortletRequestDispatcher portletRequestDispatcher =
            getPortletContext().getRequestDispatcher(path);

        if (portletRequestDispatcher == null) {
            throw new IOException(path + " is not a valid include");
        }
        else {
            portletRequestDispatcher.include(renderRequest, renderResponse);
        }
    }
}

```

Y su configuración en el fichero portlet.xml:

```

<portlet-app
  version="2.0"
  xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd http://java.sun.com/xml/ns/portlet-
>
...
  <portlet>
    <portlet-name>visualizador-parameter-portlet</portlet-name>
    <display-name>visualizador-parameter-portlet</display-name>
    <portlet-class>com.adictos.portlet.visualizador.VisualizadorParameterPortlet</portlet-class>
    <init-param>
      <name>view-jsp</name>
      <value>/view.jsp</value>
    </init-param>
    <expiration-cache>0</expiration-cache>
    <supports>
      <mime-type>text/html</mime-type>
    </supports>
    <portlet-info>
      <title>visualizador-parameter-portlet</title>
      <short-title>visualizador-parameter-portlet</short-title>
      <keywords>visualizador-parameter-portlet</keywords>
    </portlet-info>
    <security-role-ref>
      <role-name>administrator</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>guest</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>power-user</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>user</role-name>
    </security-role-ref>
    <supported-public-render-parameter>address</supported-public-render-parameter>
  </portlet>
...
  <public-render-parameter>
    <identifier>address</identifier>
    <qname xmlns:x="http://www.adictosaltrabajo.com/public-render-parameters">x:address</qname>
  </public-render-parameter>
...
</portlet-app>

```

Si desplegáis de nuevo los portlets añadiéndolos a una página, veréis que el comportamiento funcional del portlet es idéntico que el ejemplo anterior pero en este caso utilizando parámetros públicos que son compartidos por todos aquellos portlets que lo hayan registrado en su configuración.

3.3 Intercomunicación entre Portlets utilizando eventos.

La intercomunicación usando eventos se asemeja a la gestión de eventos de SWING. Existen productores de eventos y subscriptores que serán notificados cuando se lance el mismo. Los eventos sólo podrán ser lanzados en la fase de Action de un portlet utilizando el método setEvent de la clase ActionResponse. Una vez que se haya publicado un evento, el contenedor, antes de invocar a la renderización de los portlets, notificará a todos aquellos que se hubieran suscrito al evento. La configuración de productores (supported-publishing-event) y subscriptores, como os podéis imaginar, debe encontrarse en el fichero portlet.xml. A continuación se muestra la versión del ejemplo para que funcione con eventos.

Primero se crea la clase LocalizadorEventPortlet:

```

package com.adictos.portlet.localizador;

import java.io.IOException;

import javax.portlet.ActionRequest;
import javax.portlet.ActionResponse;
import javax.portlet.GenericPortlet;
import javax.portlet.PortletException;
import javax.portlet.PortletRequestDispatcher;
import javax.portlet.ProcessAction;
import javax.portlet.RenderRequest;
import javax.portlet.RenderResponse;
import javax.xml.namespace.QName;

public class LocalizadorEventPortlet extends GenericPortlet {

    protected String viewJSP;
    protected String keyAddressSession;
    public static final String KEY_JSP_ADDRESS = "address";

    @Override
    public void init() throws PortletException {
        //Se recupera la jsp encargada de pintar el portlet
        viewJSP = getInitParameter("view-jsp");
    }

    //Se define el nombre del accion que ejecutara este metodo
    @ProcessAction(name="localizar")
    public void processLocalizador(ActionRequest request, ActionResponse response) {

        //Se recupera la direccion
        String address = request.getParameter(getDefaultNamespace()+ KEY_JSP_ADDRESS);
        if (address != null) {
            //Notificamos que se ha producido un cambio en la direccion
            response.setEvent(new QName("http://www.adictosaltrabajo.com/events","addressChangeEvent"), address);
        }

    }

    @Override
    public void doView(RenderRequest request, RenderResponse response) throws IOException, PortletException {
        include(viewJSP, request, response);
    }

    protected void include(
        String path, RenderRequest renderRequest,
        RenderResponse renderResponse)
        throws IOException, PortletException {

        PortletRequestDispatcher portletRequestDispatcher =
            getPortletContext().getRequestDispatcher(path);

        if (portletRequestDispatcher == null) {
            throw new IOException(path + " is not a valid include");
        }
        else {
            portletRequestDispatcher.include(renderRequest, renderResponse);
        }
    }
}

```

Se define el portlet en el fichero portlet.xml con el evento:

```

<?xml version="1.0"?>

<portlet-app
  version="2.0"
  xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd http://java.sun.com/xml/ns/portlet:
>
...
  <portlet>
    <portlet-name>localizador-event-portlet</portlet-name>
    <display-name>localizador-event-portlet</display-name>
    <portlet-class>com.adictos.portlet.localizador.LocalizadorEventPortlet</portlet-class>
    <init-param>
      <name>view-jsp</name>
      <value>/view.jsp</value>
    </init-param>
    <expiration-cache>0</expiration-cache>
    <supports>
      <mime-type>text/html</mime-type>
    </supports>
    <portlet-info>
      <title>localizador-event-portlet</title>
      <short-title>localizador-event-portlet</short-title>
      <keywords>localizador-event-portlet</keywords>
    </portlet-info>
    <security-role-ref>
      <role-name>administrator</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>guest</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>power-user</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>user</role-name>
    </security-role-ref>
    <supported-publishing-event>
      <qname xmlns:e="http://www.adictosaltrabajo.com/events">e:addressChangeEvent</qname>
    </supported-publishing-event>
  </portlet>
  <event-definition>
    <qname xmlns:e="http://www.adictosaltrabajo.com/events">e:addressChangeEvent</qname>
    <value-type>java.lang.String</value-type>
  </event-definition>
...
</portlet-app>

```

Ahora se crea la versión del visualizador: la clase VisualizadorEventPortlet.

```

package com.adictos.portlet.visualizador;

import java.io.IOException;

import javax.portlet.EventRequest;
import javax.portlet.EventResponse;
import javax.portlet.GenericPortlet;
import javax.portlet.PortletException;
import javax.portlet.PortletRequestDispatcher;
import javax.portlet.PortletSession;
import javax.portlet.ProcessEvent;
import javax.portlet.RenderRequest;
import javax.portlet.RenderResponse;

public class VisualizadorEventPortlet extends GenericPortlet {

    protected String viewJSP;
    public static final String KEY_JSP_ADDRESS = "address";
    @Override
    public void init() throws PortletException {
        viewJSP = getInitParameter("view-jsp");
    }

    @Override
    protected void doView(RenderRequest request, RenderResponse response)
        throws PortletException, IOException {
        //Se recupera de la sesion la direccion establecida por la ejecucion del event
        PortletSession session = request.getPortletSession();
        String address = (String) session.getAttribute(KEY_JSP_ADDRESS);
        request.setAttribute(KEY_JSP_ADDRESS, address);
        include(viewJSP, request, response);
    }

    //Definimos el metodo que se ejecutara cuando se notifique el event
    @ProcessEvent(qname="{http://www.adictosaltrabajo.com/events}addressChangeEvent")
    public void onChangeAddress (EventRequest request, EventResponse response)
        throws PortletException, IOException {
        //Se recibe el evento y se establece el valor en la sesion
        PortletSession session = request.getPortletSession();
        session.setAttribute(KEY_JSP_ADDRESS, request.getEvent().getValue());
    }

    protected void include(
        String path, RenderRequest renderRequest,
        RenderResponse renderResponse)
        throws IOException, PortletException {

        PortletRequestDispatcher portletRequestDispatcher =
            getPortletContext().getRequestDispatcher(path);

        if (portletRequestDispatcher == null) {
            throw new IOException(path + " is not a valid include");
        }
        else {
            portletRequestDispatcher.include(renderRequest, renderResponse);
        }
    }
}

```

Y la correspondiente configuración en el portlet.xml:

```

<?xml version="1.0"?>

<portlet-app
  version="2.0"
  xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd http://java.sun.com/xml/ns/portlet"
>
...
  <portlet>
    <portlet-name>visualizador-event-portlet</portlet-name>
    <display-name>visualizador-event-portlet</display-name>
    <portlet-class>com.adictos.portlet.visualizador.VisualizadorEventPortle</portlet-class>
    <init-param>
      <name>view-jsp</name>
      <value>/view.jsp</value>
    </init-param>
    <expiration-cache>0</expiration-cache>
    <supports>
      <mime-type>text/html</mime-type>
    </supports>
    <portlet-info>
      <title>visualizador-event-portlet</title>
      <short-title>visualizador-event-portlet</short-title>
      <keywords>visualizador-event-portlet</keywords>
    </portlet-info>
    <security-role-ref>
      <role-name>administrator</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>guest</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>power-user</role-name>
    </security-role-ref>
    <security-role-ref>
      <role-name>user</role-name>
    </security-role-ref>
    <supported-processing-event>
      <qname xmlns:e="http://www.adictosaltrabajo.com/events">e:addressChangeEvent</qname>
    </supported-processing-event>
  </portlet>
<event-definition>
  <qname xmlns:e="http://www.adictosaltrabajo.com/events">e:addressChangeEvent</qname>
  <value-type>java.lang.String</value-type>
</event-definition>

...
</portlet-app>

```

4. Conclusión

Como se ha podido ver, el procedimiento para intercomunicar portlets con la especificación JSR 286 es muy sencillo y sistemático. Únicamente se tendrá que optar por una de las tres opciones, teniendo en cuenta siempre la más adecuada para la situación que estemos desarrollando.

[Share](#) |

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ivota!

Muy malo Malo Regular Bueno Muy bueno

Votar

(Sólo para usuarios registrados)

» [Regístrate y accede a esta y otras ventajas](#) «

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)

» [Regístrate y accede a esta y otras ventajas](#) «

Autor

Mensaje de usuario registrado

- Puedes inscribirte en nuestro servicio de notificaciones 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

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
Intercomunicación entre portlets.	Este tutorial versa sobre el uso de los nuevos métodos de intercomunicación de portlets que ofrece la especificación JSR 286 (v2 de portlets).	2010-05-17	3	-	-	
Utilización de arquetipos y plugins de LifeRay para Maven.	Uso de los nuevos arquetipos y plugins desarrollados por LifeRay para aquellos desarrolladores que utilizan Maven en su proyectos.	2010-05-10	147	-	-	
Sacar Release de un proyecto con Maven	Este tutorial muestra los pasos necesarios para sacar release de un proyecto con Maven.	2010-03-04	1601	-	-	
JavaBean Datasource Ireport	La particularidad del caso que nos ocupa, es conseguir que la fuente de datos del informe sea una lista de JavaBeans y no una consulta definida previamente en el informe.	2009-12-14	2920	Bueno	1	
Maven Assembly Plugin: empaquetando aplicaciones con Maven para la ejecución de procesos batch.	Examinamos cómo llevar a cabo el empaquetado de nuestras utilidades batch, basadas en una capa de servicios y cómo hacer la distribución independiente del entorno.	2009-11-12	2097	-	-	
Cómo conseguir que Subversion avise a Hudson para lanzar una build	En este tutorial vamos a ver como configurar Subversion para que sea este el que avise a Hudson cada vez que hay un commit, y así se lance la build.	2009-10-27	3460	-	-	
Cómo instalar Hudson en Apache Tomcat	Instalar Hudson en Apache Tomcat	2009-10-26	4101	Muy bueno	1	
Mavenizar Liferay SDK	En este tutorial vamos a ver unos sencillos pasos para poder hacer plugins de tipo portlet para Liferay adaptando el entorno de desarrollo Liferay Plugins SDK	2009-09-27	1763	Regular	1	
EJB 3.0 y pruebas unitarias con Maven, JUnit 4 y Apache Open EJB.	Continuamos buscando una buena solución para llevar a cabo test de EJBs, ahora con OpenEJB.	2009-09-23	3482	-	-	
EJB 3.0 y pruebas de persistencia con Maven, JUnit 4 y Embedded JBoss sobre Java 6.	Damos continuidad al tutorial EJB 3.0 y pruebas unitarias con Maven, JUnit 4 y Embedded JBoss Java 6, probando el llevar a cabo un test de persistencia de un EJB de entidad, bajo soporte de EJB de servicio que implementa el patrón dao, en el mismo entorno	2009-09-14	3836	Muy bueno	1	

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.