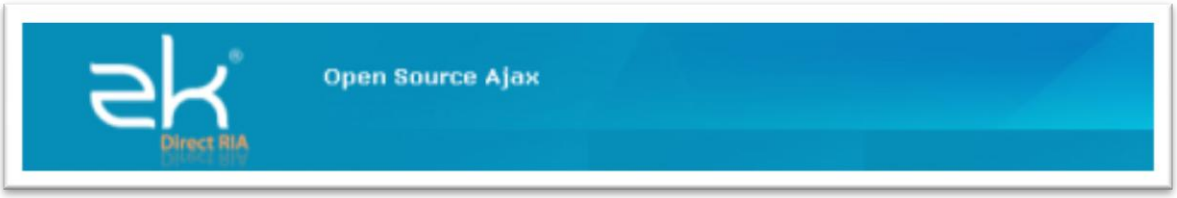

Contenido

Introducción

Este artículo describe la integración de Applets y Zk con el objeto de maximizar el uso de AWT y Swing en aplicaciones Webs. Se considerará la comunicación bidireccional entre Applets y Zk, la utilización de firma digital en Applets y la solución de problemas de visualización de ventanas emergentes sobre Applets.

Comunicación Bidireccional

Tener alguna idea acerca de qué son los Applets y cómo pueden mejorar las aplicaciones web a través de funcionalidades complejas y módulos creados con Swing o AWT parece ser de conocimiento común. Sin embargo, la comunicación defectuosa entre cliente y servidor (que imposibilita la comunicación de eventos producidos en cualquiera de los dos lados) lleva a que la flexibilidad que aportan Swing o AWT en la construcción de funcionalidad compleja se vea desfavorecida cuando se intenta agregar un Applet a la aplicación web.

Afortunadamente, existe una forma simple de generar una comunicación bidireccional básica que utiliza Javascript como medio de integración para notificar al server Zk por medio de eventos generados con ZK AU.

El concepto puede está ilustrado en la imagen 1:

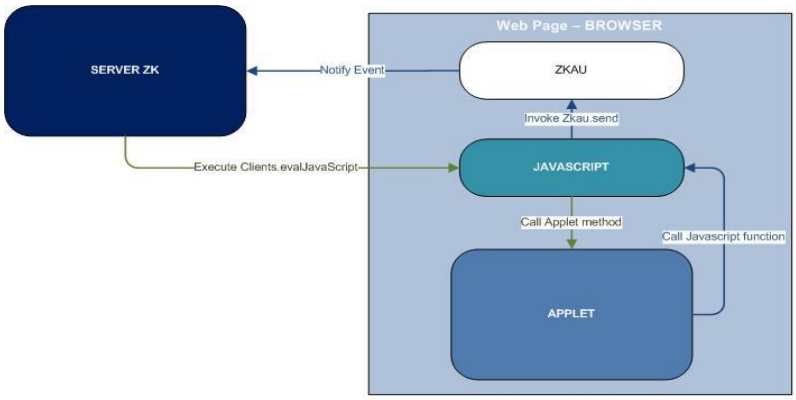

Imagen 1

Para poder llevar a cabo la comunicación es preciso seguir los pasos detallados a continuación:

1- Definir un componente HTML y construir un Applet básico

Existen dos formas de incorporar un Applet a una página Zk. La primera y más sencilla consiste en utilizar el componente Applet proporcionado por Zk en las últimas versiones y completar los parámetros requeridos. La segunda y más compleja se consigue a través de la definición de un componente HTML que contenga al Applet tag construido de forma tradicional. Para este SmallTalk, se utiliza la última opción con objeto de obtener retro compatibilidad con versiones antiguas de Zk, donde todavía no existía el componente Applet.

La estructura de la entrada del ZUL debe ser como la siguiente:

```
<html id="appletVisor" width="100%" height="100%" visible="false"/>
```

En el método de inicialización de la página se debe definir el contenido del Applet:

```
String appletDef= "<applet id=\"viewerApplet\"/></applet>";

Html html= (Html)this.getFellow("appletVisor");
html.setContent(appletDef);
```

Después de efectuar esta acción, se habilita el segundo paso.

2- Crear eventos personalizados del lado del servidor

Para generar eventos personalizados a través de la invocación de ZK AU, se debe definir cómo van a ser interpretados por el servidor. En primer lugar, hay que determinar el método ***getCommand*** para procesar cada comando y darles el tratamiento adecuado acorde a las necesidades del usuario:

```
public class TestUI extends Window implements EventListener{

}
```

```
public Command getCommand(String cmdId) {
         if(cmdId.equals("notifyServer")) {
            return new DisplayCommand(cmdId);
         }
   return super.getCommand(cmdId);
}
```

A continuación, necesitamos construir nuestra propia versión de ***ComponentCommand*** para que interprete el comando y genere un evento personalizado:

```
private class DisplayCommand extends ComponentCommand {

public DisplayCommand(String command) {
super(command, Command.SKIP_IF_EVER_ERROR|Command.CTRL_GROUP);
        }

        protected void process(AuRequest request) {
Events.postEvent(new
Event(request.getCommand().getId(),request.getComponent(),request.getD
ata()));
        }
}
```

En el método de inicialización se debe registrar el evento recién creado y habilitar el Listener que se encargará de atenderlo cuando se dispare:

```
this.addEventListener("onDocumentSaved", this);
```

Por último, se debe habilitar el método ***onEvent***:

```
public void onEvent(Event evt){
if(evt.getName().equals("notifyServer")){

}

}
```

Así se completa el ciclo de notificación del lado del Servidor.

3- Crear métodos en el Applet que atiendan e invoquen los eventos

A continuación, se debe construir un Applet en Java de la manera tradicional e incorporar dos métodos, uno para procesar una llamada del Servidor y otro para generar una llamada al mismo.

a) Recibir una llamada del servidor

```
public void notifyApplet() throws IOException {

//do something
}
```

b) Generar un evento al Servidor

- Primero, se debe habilitar Javascript en nuestro Applet a través de la inscripción de `MAYSCRIPT>` al final del tag.

- Utilizar **JSObject** para realizar la llamada Javascript. Es necesario que plugin-x.x.x.jar esté dentro de nuestro classpath para poder compilar el Applet sin errores. En el ejemplo, utilizamos plugin-1.4.2.jar correspondiente a la JDK 1.4.2:

```
public void notifyServer(){
//do something

JSObject jso = JSObject.getWindow(this.applet);

            jso.call("notifyServer", new Object[] { uuid });

}
```

4- Crear funciones Javascript en el ZUL

Para culminar la integración, se necesita definir en la página una función Javascript cuyo nombre sea igual al definido en la llamada en **JSObject**:

```
      <script type="text/javascript">
<![CDATA[
function notifyServer (uid, text){
zkau.send({uuid:uid,cmd:"notifyServer",data:[text],ctl:true});
]]>
      </script>
```

5- Invocar un método del Applet utilizando Javascript

El método del Applet se invoca de la siguiente manera:

```
Clients.evalJavaScript("document.getElementById('appletVisor').notifyA
pplet()");              }
```

Luego de haber completado cada uno de los pasos arriba descriptos, podemos graficar la secuencia de comunicación de la siguiente manera:

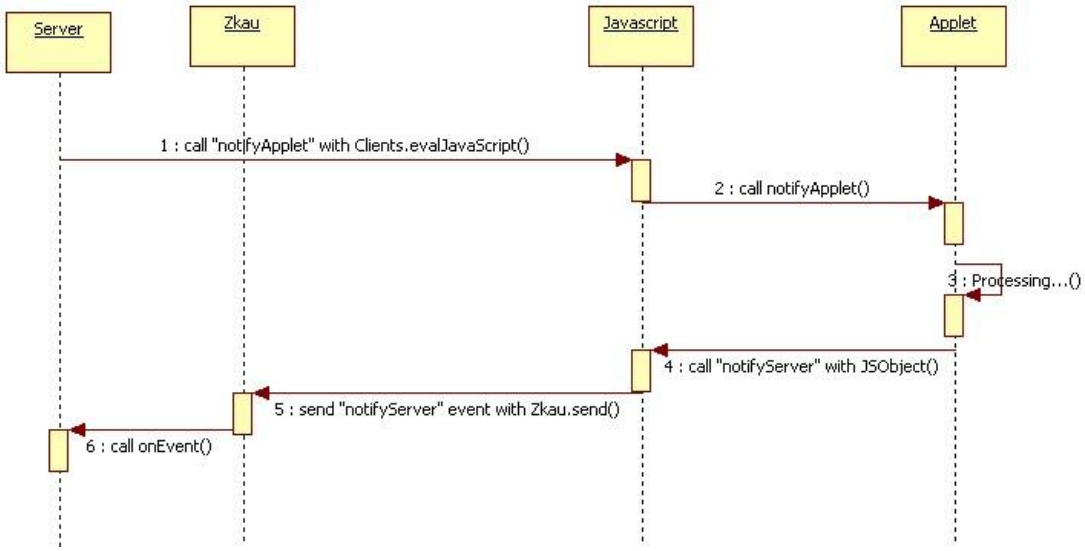

Imagen 2

Ventanas modales y Applets

Existe un punto más a considerar cuando integramos Applets dentro de nuestras aplicaciones web ZK. Al utilizar ventanas modales sobre Iframes o Applets, se produce su ocultamiento/ solapamiento, lo que las hace inaccesibles y bloquea cualquier otra función de la aplicación al quedar en estado modal.

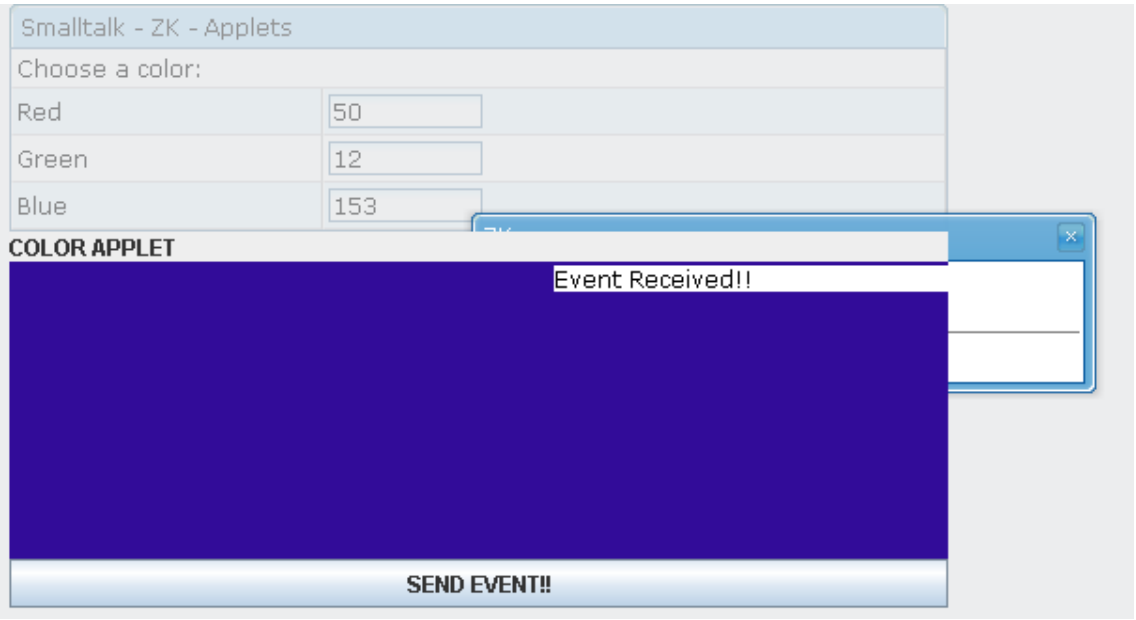

Imagen 3

Para corregir el problema y dependiendo de la versión de Zk que se esté utilizando, se debe incorporar:

Versiones de Zk 3.0.x: debe incorporarse la propiedad authide con el objeto de ocultar el Applet o Iframe mientras exista una ventana modal por delante. Se debe agregar:

```
<script type="text/javascript">zk._actg1 =
["IFRAME","EMBED","APPLET"];</script>
```

Y luego en nuestro applet colocar la propiedad *autohide*:

```
<applet id="testApplet" z.autohide="true" . . ./>
```

Versión de Zk 3.6.1: la aplicación lo maneja automáticamente y no hay que realizar agregados extras para que puedan visualizarse las ventanas emergentes.

Versión de Zk 3.6.2: debido a problemas de performance, se desactivó la funcionalidad incorporada en la versión anterior. Para activarla manualmente se debe agregar:

```
<?script content="zk.useStackup = true;"?>
```

Mientras la Variable Global este activa, las ventanas emergentes se renderizarán por delante de los Applets / Iframes.

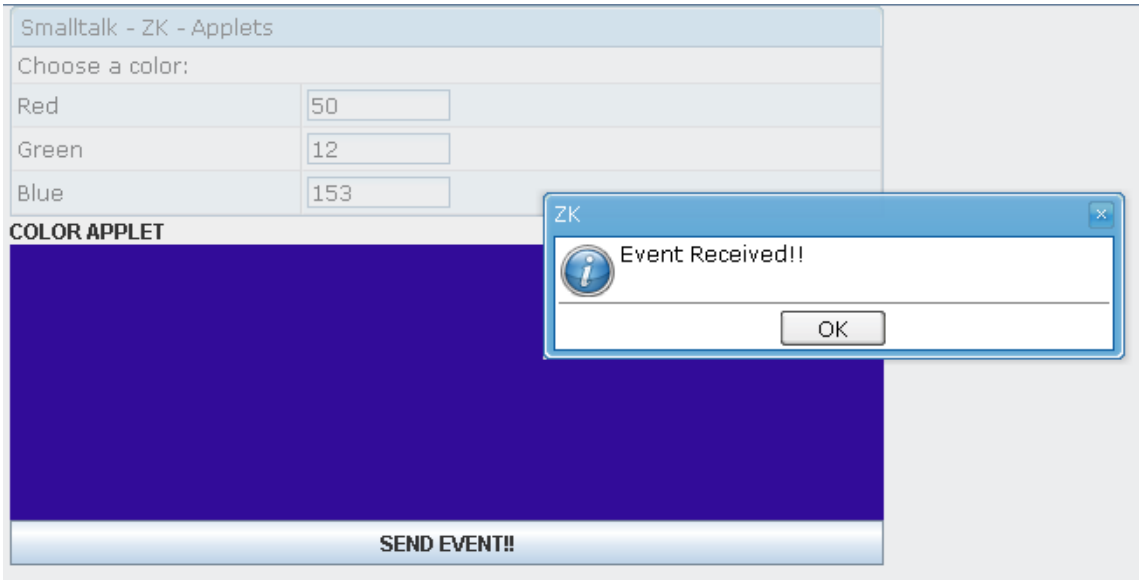

Imagen 4

Resumen

La integración de Applets con Zk permite maximizar las posibilidades de desarrollo en esta plataforma: aporta una forma sencilla para incorporar funcionalidad desktop compleja dentro de una página Web sin perder la comunicación cliente – servidor.

Autores

Pablo H. Giménez
Juan P. Francisconi