

Flash y java mediante Web Services

Daniel Monedero Tórtola

faltantornillos@gmail.com

<http://www.faltantornillos.net>

Índice de contenido

Licencia.....	3
Introducción.....	4
Java.....	5
Flash.....	6
Conclusión.....	7

Licencia

Estos textos se distribuyen bajo licencia Creative Commons Reconocimiento-NoComercial- CompartirIgual 2.5 España (<http://creativecommons.org/licenses/by-nc-sa/2.5/es/>).

Introducción

Los Web Services o servicios webs son una forma de comunicar aplicaciones diferentes usando XML. Esto resulta muy útil ya que nos brinda una opción de comunicación estándar que podemos usar entre aplicaciones realizadas con distintos lenguajes como actionscript, c# o java.

Para este tutorial en concreto vamos a ver cómo podemos comunicar aplicaciones actionscript (realizadas con Macromedia Flash) y aplicaciones java (realizadas con Jbuilder), en cuyo caso se usa SOAP, un protocolo de acceso a objetos basado en XML, aunque el uso de XML va a ser completamente transparente al programador y no nos debe preocupar ese aspecto.

Java

Lo primero que hay que hacer es crear un proyecto con un método en java, este método será llamado por flash mediante webservices, puede hacer cualquier cosa que se pueda hacer en java pero si envía o recibe parámetros estos deben ser serializables, es decir, tienen que ser objetos que contengan cadenas o vectores con cadenas (o vectores de vectores con cadenas, etc) pero no estructuras más complejas como tablas hash, árboles, etc.

Una vez creado el método en JBuilder hacemos click derecho sobre él y seleccionamos "Export as a Web Service", si no tenemos ningún otro Web Service en el proyecto o queremos que se ejecute en uno aparte seleccionamos "Create web services module", en web module seleccionamos "New", ahora seleccionamos "Single server for all services in project" y elegimos Tomcat 4.1 o 5 según la versión que vayamos a utilizar. Finish. Finish. Ahora nos aparecerá el Web Services Designer donde podemos ver gráficamente nuestro Web Service. Para lanzarlo: botón derecho en la clase y pulsamos sobre "Rebuild", una vez ha acabado pulsamos sobre "Run Project" (un botón similar al de play pero en verde). En este momento ya tenemos Tomcat y nuestro Web Service ejecutándose, solo queda diseñar la parte de flash.

Flash

Abrimos Macromedia Flash y seleccionamos "Archivo", "Nuevo", "Documento de Flash" o abrimos un archivo flash que queramos conectar con nuestra función java. Ahora, en la derecha, en el panel de "Componentes" arrastramos el componente "WebServiceConnector" a nuestra área de trabajo. Para poner a punto el Web Service Connector le damos un nombre (pestaña "Propiedades", label "Componente") por ejemplo WSC, introducimos la WSDLURL (suele ser una url del tipo <http://10.142.85.97:8080/WebModule1/services/pruebaWS?wsdl>, la podemos consultar mediante la ventana de navegación que se nos ha abierto en el JBuilder) y el nombre del método al que queremos llamar, por ejemplo enviar cadena. Todos estos campos se pueden cambiar en tiempo de ejecución mediante `_root.WSC`.

Para activar el Web Service podemos hacerlo cuando se carga el archivo swf, cuando se pulsa un botón, etc. Lo importante es poner el siguiente código para enviar y recibir los parámetros:

```
//Para el listener
var wscListener:Object = new Object();
wscListener.result = function(evt:Object) {
    //en evt.target.results tenemos las variables devueltas por el método java
    trace("Cadena obtenida: " + evt.target.results);
};
//Anyadimos el listener
_root.WSC.addEventListener("result", wscListener);
//en _root.WSC.params introducimos los datos que le llegan al método java
_root.WSC.params=["Te envio esta cadena"];
//que no se nos olvide activar el Web Service! :D
_root.WSC.trigger();
```

Conclusión

Y eso es todo, a pesar de la simplicidad del ejemplo los Web Services son una herramienta muy potente que con el tiempo substituirá a otras interfaces entre aplicaciones. Además, el hecho de que esté basado en XML mediante SOAP nos asegura la compatibilidad con cualquier lenguaje que soporte dicho protocolo.