

Cliente Axis2 – Emi Kohmann

Antes de hacer esto, recordar tener levantado el servicio Axis2.

1. **File >> New >> Web Dynamic Project.** Ponemos el nombre del proyecto (EjemploAxisCliente)
2. En la misma ventana vamos a **Configuration >> Modify**
3. Tildamos **Axis2 Web Services** y bajamos **Dynamic Web Module** a **2.5**
4. **Next >> Next >>** Dejamos tildado "**Generate web.xml deployment descriptor**"
5. **Finish.** Ahora ya tenemos el proyecto y tenemos que crear el servlet que lo consuma.
6. Creamos el paquete **ar.edu.ubp.das.servlets**
7. **Click derecho en el paquete >> New >> Servlet >> EjemploClienteServlet**
8. Dejamos todo como está. Ahora podemos ver que (a diferencia de otros proyectos en Eclipse) el servlet no tiene la notation `index.jsp`, porque estamos usando un Dynamic Web Module anterior. No es necesario que programemos nada todavía.
9. **Click derecho en el proyecto >> New >> Other >> Web Service Client.**
10. En **Service Definition** pegamos el **link de la wsdl** que obtuvimos en el otro instructivo. Sólo nos dejará continuar siempre y cuando el servicio esté levantado y la url sea correcta.
11. **Configuration >> Web Service runtime >>** Cambiamos **Apache Axis** por **Apache Axis 2.**
12. **Next >>** Dejamos todo como está >> **Finish.**
13. Esto nos generó el paquete **ws** que contiene el **CallbackHandler** y el **Stub**. Estas clases son las que vamos a usar en el Servlet para consumir Axis2.
14. Listo, el servicio se consume de la siguiente manera. **Todas las clases utilizadas a partir de ahora en más, son las que nos generó Eclipse. Es importante que escribamos el nombre y después hagamos Ctrl + Shift + O para que se incluyan, sino va a tirar error.**

15. En la función **doGet()**, o donde nosotros querramos, creamos un objeto para el servicio y un objeto para manejar la función (generalmente esto es **NombreDelServicio** seguido de **Stub**).

```
EjemploStub stub = new EjemploStub();
```

16. Luego creamos un objeto para manejar la función que programamos en el servicio.

```
FuncionEjemplo req = new FuncionEjemplo();
```

17. Opcional: Si la función recibe parámetros, lo enviamos mediante:

```
req.setNombreParametro( parametro );
```

18. Luego obtenemos la respuesta creando un objeto de la clase que se llama como la función seguido de **Response**, y usamos la función **get_return()**.

```
FuncionEjemploResponse resp = stub.funcionEjemplo(req);  
String respuesta = resp.get_return();  
response.getWriter().println( respuesta );
```

19. **Run >> Run on server** y efectivamente consumió.

localhost:8080/3.PruebaAxisCliente/index.jsp

hola