

Servicio Web con CXF – Emi Kohmann

1. **New >> Dynamic Web Project.** Nombre del proyecto (PruebaCXF).
2. En **Configuration** tenemos que ir a **Modify** y tildar la opción **CXF 2.x Web Services**.
3. **Ok >> Next >> Next >>** Acá debemos tildar **Generate web.xml**.
4. **Next >>** Acá si está bien configurado, ya debería salirnos **Apache CXF 3.1.7 >> Finish**.
5. Bien, ya tenemos el proyecto y ahora debemos crear la clase que implementa el servicio.
6. **Click derecho en src >> New >> Package >> ar.edu.ubp.das.ws**
7. **Click derecho en el paquete >> New >> Class >> Ejemplo** (o lo que sea, es una clase vacía).
8. Lo importante ahora es programar el comportamiento antes de agregar el servicio web. Es decir, vamos a programar la función de ejemplo y luego vamos a seguir.
9. Apenas programemos esto, vamos a tener una clase simple como esta.
10. **Click derecho sobre la clase >> New >> Other >> Web Service.**
11. Vamos a **Configuration >> Web Service runtime**.
12. Modificamos **Apache Axis** por **Apache CXF 2.x**.
13. **Ok >> Next >>** Acá si la tenés clara activa la interfaz, sino seguí >> **Next**.
14. Acá tenemos la vista previa de cómo quedaría la clase >> Dejamos todo como está >> **Next**
15. Antes de continuar, es importante que tildemos **Generate server** para que nos genere la **JavaApplication**.
16. Dejamos todo como está >> **Next >>** Y acá se puede demorar un rato hasta que levanta.

```
package ar.edu.ubp.das.ws;  
  
public class Ejemplo {  
  
    public String funcionEjemplo() {  
  
        return "hola cxf";  
    }  
}
```

Importante: Acá podríamos tener algún error del servidor, ver cómo solucionar los problemas para que arranque. Si hubo algún problema con el servidor mientras esto se creaba, los paquetes y las clases van a quedar incompletas. Es necesario **pisar** todo lo que se creó volviendo al paso 10. Esto va a pisar todo lo incompleto y va a funcionar.

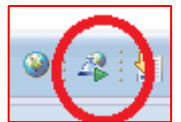
17. Okey, una vez que levantó todo, nos debería haber generado 2 paquetes: **ws** (donde ya teníamos nuestra clase que ahora está modificada y nos generó un **PortTypeServer**) y **jaxws** (con las clases para consumir el servicio).
18. Ahora vamos a ir a la **Java Application** (es decir, al **Ejemplo_PortTypeServer.java**) y vamos a comentar las siguientes líneas:

```
// Thread.sleep(5 * 60 * 1000);  
// System.out.println("Server exiting");  
// System.exit(0);
```

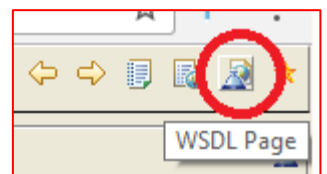
Esto es para que no se nos caiga a los 5 minutos. Esta clase es una aplicación de Java que levanta el servidor. Entonces le vamos a dar **click derecho >> Run as >> Java Application**. Debería decir **Server ready...**

19. Ahora vamos a copiar la dirección que está en **String address**, en esta misma clase.
`String address = "http://localhost:9090/EjemploPort";`
20. Y le agregamos al final **?wsdl**. Accedemos en el navegador (**No** hacer run on server) entonces a: <http://localhost:9090/EjemploPort?wsdl>. Ahí vamos a ver un XML.

21. Ahora vamos a probar si funciona el servicio. Es importante **no correr este proyecto como Run As >> Run on Server**, porque nos podría complicar después. Vamos a usar este ícono para ver si funciona el servicio. Este ícono está arriba a la derecha en la barra de herramientas.



22. Una vez que se nos abre la página del explorador de servicios, **debemos apretar en este botón para no renegar** (si no buscamos este link no anda).



23. Ese link nos lleva a la **WSDL Page**. Allí podremos pegar el link de nuestra wsdl, es decir <http://localhost:9090/EjemploPort?wsdl>, seleccionamos **WSDL main**, pegamos la url en **WSDL url**, click en **go** y nos debería figurar nuestra función. Hacemos **click sobre la función** (agregamos parámetros si es necesario) y en la sección **status** ya nos tiene que devolver el resultado de consumir el servicio desde el explorador.

