

## CREAR CLIENTE SOAP CON JAVA

(wsimport - Java™ API for XML Web Services (JAX-WS) 2.0)

```
C:\>wsimport -keep -verbose http://localhost:9999/ws/mensaje?wsdl
```

## AUTORIZACION

```
http://user:password@host:port//url-path
```

---

CREAR WEB SERVICE JAX-WS con java

CREAMOS INTERFACE DEL SERVICIO WEB

```
package servweb;
import javax.jws.WebMethod;
import javax.jws.WebService;
import javax.jws.soap.SOAPBinding;
import javax.jws.soap.SOAPBinding.Style;
//Los import se usan con las anotaciones de abajo.
@WebService
@SOAPBinding(style = Style.RPC)

public interface Mensaje {
    //para crear un metodo de un web service
    @WebMethod String metodoWeb(String cadenaDeCaracteres);
}
```

CREAMOS LA CLASE LA CUAL IMPLEMENTA LA INTERFACE

```
package servweb;

import javax.jws.WebService;

//Implementacion del servicio
@WebService(endpointInterface = "servweb.Mensaje")
public class ServWeb implements Mensaje{
    //aqui implementa la interface
    @Override
    public String metodoWeb(String cadenaDeCaracteres) {
        return "Mensaje: " + cadenaDeCaracteres;
    }
}
```

```
}
```

CREAMOS LA CLASE QUE LA PUBLICA EN LA URL

```
package servweb;

import javax.xml.ws.Endpoint;
//importamos el servweb
import servweb.ServWeb;
//corre el web service
public class publisher {
    public static void main(String[] args) {
        Endpoint.publish("http://localhost:9999/ws/mensaje", new ServWeb());
    }
}
```

PROBAMOS CON LA URL:

<http://localhost:9999/ws/mensaje?wsdl>



```
<!--
Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e946065a4c2f4e30e065b245e51e.
-->
<!--
Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e946065a4c2f4e30e065b245e51e.
-->
<definitions targetNamespace="http://servweb/" name="ServWebService">
  <types/>
  <message name="metodoWeb">
    <part name="arg0" type="xsd:string"/>
  </message>
  <message name="metodoWebResponse">
    <part name="return" type="xsd:string"/>
  </message>
  <portType name="Mensaje">
    <operation name="metodoWeb">
      <input wsam:Action="http://servweb/Mensaje/metodoWebRequest" message="tns:metodoWeb"/>
      <output wsam:Action="http://servweb/Mensaje/metodoWebResponse" message="tns:metodoWebResponse"/>
    </operation>
  </portType>
  <binding name="ServWebPortBinding" type="tns:Mensaje">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
    <operation name="metodoWeb">
      <soap:operation soapAction="">
        <input>
          <soap:body use="literal" namespace="http://servweb/">
        </input>
        <output>
          <soap:body use="literal" namespace="http://servweb/">
        </output>
      </operation>
    </binding>
  </service name="ServWebService">
    <port name="ServWebPort" binding="tns:ServWebPortBinding">
      <soap:address location="http://localhost:9999/ws/mensaje"/>
    </port>
  </service>
</definitions>
```

CREAR CLIENTE JAVA PARA EL WEB SERVICE

CREAMOS CLASE CLIENTE

```

package servweb;
import java.net.URL;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;
//importamos la interface
import servweb.Mensaje;
public class cliente {
    public static void main(String[] args) throws Exception {

        URL url = new URL("http://localhost:9999/ws/mensaje?wsdl");

        //1st argument service URI, refer to wsdl document above
        //2nd argument is service name, refer to wsdl document above
        QName qname = new QName("http://servweb/", "ServWebService");

        Service service = Service.create(url, qname);

        Mensaje hello = service.getPort(Mensaje.class);

        System.out.println(hello.metodoWeb("Hecho!"));

    }
}

```



## CLIENTE CON WSIMPORT TOOL

### USAMOS LA HERRAMIENTA WSIMPORT TOOL

```
wsimport -keep http://localhost:9999/ws/mensaje?wsdl
```

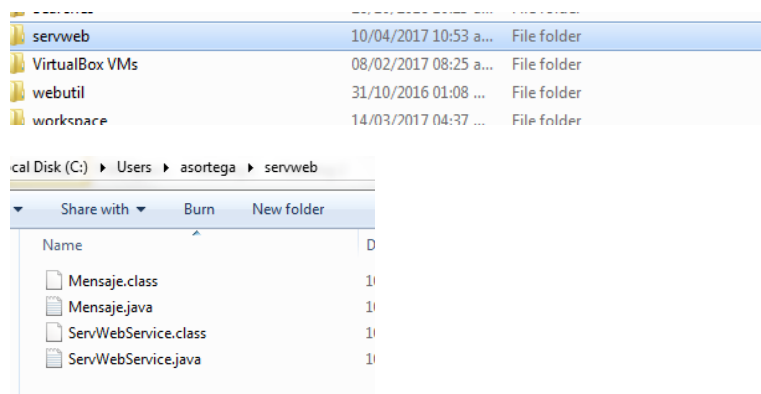
```
C:\Users\asortega>wsimport -keep http://localhost:9999/ws/mensaje?wsdl
parsing WSDL...

Generating code...

Compiling code...

C:\Users\asortega>
```

EN LA DIRECCION USADA CREA UNA CARPETA CON LOS ARCHIVOS CREADOS.



CREAMOS UNA CLASE QUE DEPENDA DE LOS ARCHIVOS GENERADOS

```
package servcli;

public class ServCli {

    public static void main(String[] args) {
        //se crea objeto del servicio web
        ServWebService servicio = new ServWebService();
        //se busca el metodo del puerto
        Mensaje mensaje = servicio.getServWebPort();

        System.out.println(mensaje.metodoWeb("Hecho!"));
    }
}
```

(En este ejemplo los metimos al mismo paquete, para ni importarlos)

“Mensaje: “ lo regresa desde el web service

“Hecho!” lo regresa según el input del cliente.