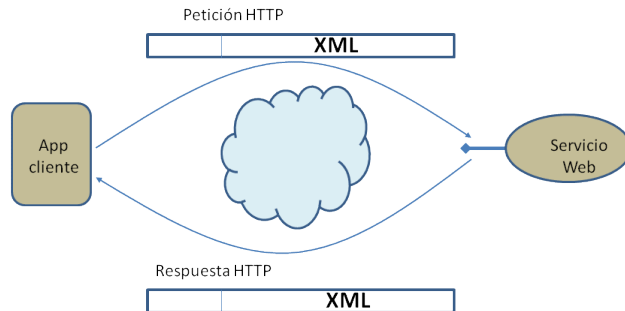


SERVICIOS WEB XML



Características y funcionamiento

Como ya hemos indicado, un servicio Web XML es aquel que expone una serie de operaciones en la Web a las que se accede mediante el intercambio de documentos XML entre el servicio y la aplicación cliente. Estos **documentos deben tener un formato específico**, definido por uno de los estándares de la arquitectura de servicios Web XML



Debido a su rigidez y peor eficiencia respecto a REST, esta aproximación es cada vez menos usada y, aunque no la vamos a tratar en profundidad en este curso, conviene conocer los diferentes estándares y tecnologías en las que se basa

Estándares

En los servicios web XML, el uso de HTTP como protocolo de comunicación y de XML como formato de intercambio de datos constituyen la base de la arquitectura. Pero, además de estos estándares básicos, hay ciertos problemas que resolver cuando se van a implementar aplicaciones basadas en servicios web, como la **estructura que tienen que tener los documentos XML** que se van a intercambiar el servicio web con sus clientes, o la manera en la que se deben especificar las **operaciones que un servicio web ofrece**

Para la resolución de estas cuestiones, el Consorcio Internacional de la Web (**W3C**) ha **definido unos estándares**, que establecen la forma en la que un servicio web debe definir su interfaz y cómo debe interaccionar con las aplicaciones clientes. Estos estándares son:

- **SOAP**. *Simple Object Access Protocol*, establece el **formato de los documentos XML** que deben intercambiarse el servicio web y las aplicaciones clientes, tanto durante la llamada a los métodos del servicio como al generar la respuesta.
- **WSDL**. *Web Service Description Language* también es un lenguaje basado en XML y se emplea para **describir la interfaz del servicio**, es decir, qué operaciones expone, parámetros de entrada y salida, así como la localización del propio servicio

El estándar SOAP

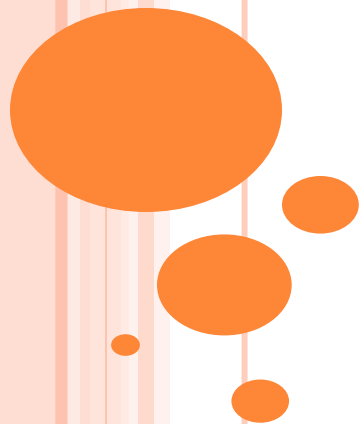
Comentemos brevemente los fundamentos de *Simple Object Application Protocol* (SOAP). Se trata de un lenguaje XML que se emplea para **definir la estructura y formato de los documentos XML que se van a intercambiar el servicio web con las aplicaciones clientes**. Estos documentos viajan en el cuerpo de las peticiones y respuestas HTTP, y se les conoce como **mensajes SOAP**.

Un mensaje SOAP incluye:

- **Envelope**. Se trata del elemento raíz de un documento SOAP. Define las **referencias a los distintos espacios de nombres utilizados por el documento**..
- **Header**. Es un elemento opcional que contiene **información de control** sobre el documento.
- **Body**. Es el elemento más importante de un documento SOAP, ya que contiene la parte principal del mensaje SOAP. Su estructura es muy sencilla; en el caso de los mensajes de llamada a una operación, el interior del *body* incluirá un elemento con el nombre de la operación a invocar y, dentro de este, un grupo de subelementos con los argumentos de la llamada. En el mensaje de respuesta, el *body* incluirá tan solo un elemento con el resultado devuelto por el método

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns:comprobarPrecio xmlns:ns="http://MyServices/">
      <cod>B7AW</cod>
    </ns:comprobarStock>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

WSDL



El estándar WSDL

Para que un cliente pueda interactuar con un servicio Web es necesario que este esté descrito de alguna manera. WSDL o Lenguaje de Descripción de Servicios Web es una especificación basada en XML, utilizada para la descripción de servicios web.

Los proveedores de servicios web utilizan WSDL para generar un documento XML en el que describen las características del servicio que ofrecen, los clientes del servicio web utilizan este documento para obtener la información necesaria que les permita utilizarlo.

Normalmente no será necesario generar este documento a mano, muchas de las herramientas existentes para la creación de servicios web son capaces de generar directamente el documento WSDL correspondiente.

Más concretamente, un documento WSDL describe:

- Los métodos proporcionados por el servicio
- Los parámetros de entrada y salida
- Como conectarse con el servicio



Ejemplo documento WSDL

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="http://com.test/wsdl/Cuenta"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  targetNamespace="http://com.test/wsdl/Cuenta">
  <types>
    <xsd:schema targetNamespace=" http://com.test/wsdl/Cuenta">
      <xsd:element name="GetPrecio">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element minOccurs="0" maxOccurs="1" name="codigo" type="xsd:string" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="GetPrecioResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element minOccurs="1" maxOccurs="1" name="Precio" type="xsd:int" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>

  <message name="GetPrecioSoapIn">
    <part name="parameters" element="tns:GetPrecio" />
  </message>
  <message name="GetPrecioSoapOut">
    <part name="parameters" element="tns:GetPrecioResponse" />
  </message>

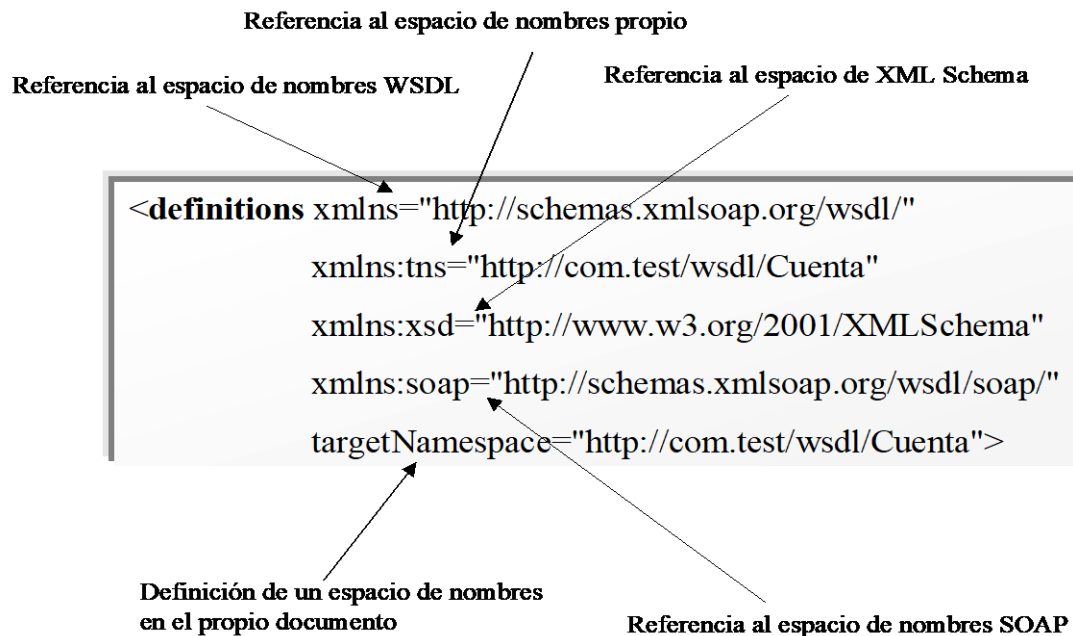
  <portType name="Service1Soap">
    <operation name="GetPrecio">
      <input message="tns:GetPrecioSoapIn" />
      <output message="tns:GetPrecioSoapOut" />
    </operation>
  </portType>

  <binding name="Service1SoapBinding" type="tns:Service1Soap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
    <operation name="GetPrecio">
      <soap:operation soapAction=" http://com.test/wsdl/Cuenta /GetPrecio" style="document" />
      <input>
        <soap:body use="literal" />
      </input>
      <output>
        <soap:body use="literal" />
      </output>
    </operation>
  </binding>
</definitions>
```



definitions

Todo documento WSDL está formado por un elemento raíz llamado **definitions**, en él se declaran los diferentes espacios de nombres utilizados en el documento



types

Se utiliza para definir los tipos de datos que van a ser utilizados por el servicio, tanto en lo que se refiere a parámetros de métodos como a valores devueltos por estos

Elementos

```
<types>
  <xsd:schema targetNamespace=" http://com.test/wsdl/Cuenta">
    <xsd:element name="GetPrecio">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element minOccurs="0" maxOccurs="1" name="codigo" type="xsd:string" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="GetPrecioResponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element minOccurs="1" maxOccurs="1" name="Precio" type="xsd:int" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</types>
```



message

Los elementos **message** representan la información (mensajes) que va a ser intercambiada entre los métodos del servicio Web y los clientes. Cada *message* tiene un subelemento **part** que describe el tipo de datos utilizado por el mensaje

Mensaje que utiliza el tipo de dato GetPrecio

```
<message name="GetPrecioSoapIn">
  <part name="parameters" element="tns:GetPrecio" />
</message>

<message name="GetPrecioSoapOut">
  <part name="parameters" element="tns:GetPrecioResponse" />
</message>
```

Mensaje que utiliza el tipo de dato GetPrecioResponse



portType

Describe las operaciones expuestas por el servicio Web, cada operación está definida con el elemento ***operation*** y llevará asociado un mensaje de entrada y otro de salida

Método expuesto por el servicio

```
<portType name="Service1Soap">
  <operation name="GetPrecio">
    <input message="tns:GetPrecioSoapIn" />
    <output message="tns:GetPrecioSoapOut" />
  </operation>
</portType>
```



binding

Una vez definidas las operaciones que va a realizar el servicio es necesario vincularlas a un protocolo concreto. Esta es la labor realizada por el elemento binding, en donde se especifica el tipo de transporte utilizado (soap, http get, http post, etc.) y los accesos de red necesarios para ejecutar una determinada operación



service

Finalmente, es necesario especificar la dirección o punto de entrada al servicio Web, información incluida dentro del elemento `service`.

Mediante el subelemento ***part*** se especifica la dirección de cada elemento *binding*. En el caso de soap, esta dirección es indicada con el atributo ***location*** del elemento ***address***

Localización del servicio Web

binding

```
<service name="Primero">
  <port name="ServicePrimero" binding="tns:Service1SoapBinding">
    <soap:address location="http://localhost:8080/primero-jaxrpc/primero"/>
  </port>
</service>
```