



SNS COLLEGE OF TECHNOLOGY, COIMBATORE-35

(AN AUTONOMOUS INSTITUTION)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

19CSO303-Web Designing(Open Elective)

UNIT-V

WEB SERVICES

Introduction to Web service -Transport protocols for web services – messaging with web services - protocols - SOAP - describing web services – WSDL – Anatomy of WSDL – manipulating WSDL – web service policy – Case Study : Web Hosting(By FTP) and Front End Development Tools –Web API

Introduction to Web service:

- A web service is any piece of software that makes itself available over the internet and uses a standardized XML messaging system.
- XML is used to encode all communications to a web service.
- For example, a client invokes a web service by sending an XML message, then waits for a corresponding XML response. As all communication is in XML, web services are not tied to any one operating system or programming language—Java can talk with Perl; Windows applications can talk with Unix applications.
- A web service is a collection of open protocols and standards used for exchanging data between applications or systems.
- Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. This interoperability (e.g., between Java and Python, or Windows and Linux applications) is due to the use of open standards.

Components of Web Services

The basic web services platform is XML + HTTP. All the standard web services work using the following components –

- SOAP (Simple Object Access Protocol)

- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)

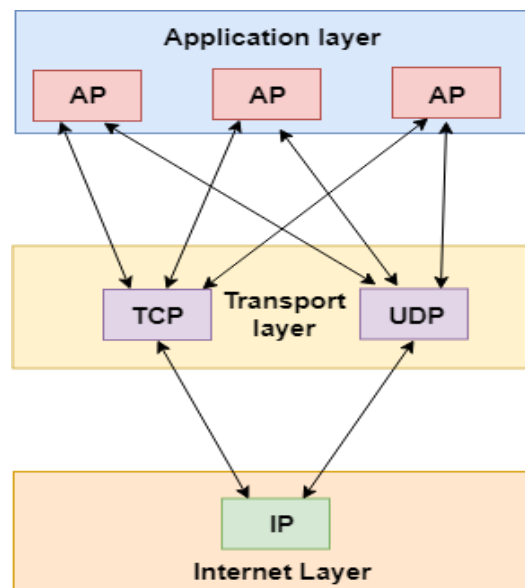
The steps to perform this operation are as follows –

- The client program bundles the account registration information into a SOAP message.
- This SOAP message is sent to the web service as the body of an HTTP POST request.
- The web service unpacks the SOAP request and converts it into a command that the application can understand.
- The application processes the information as required and responds with a new unique account number for that customer.
- Next, the web service packages the response into another SOAP message, which it sends back to the client program in response to its HTTP request.
- The client program unpacks the SOAP message to obtain the results of the account registration process.

Transport protocols for web services:

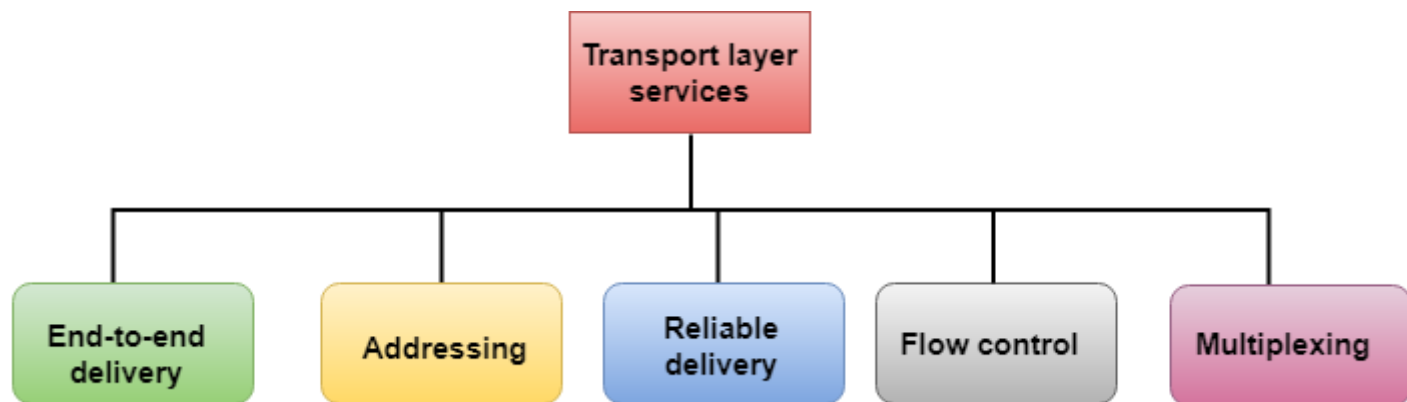
- The transport layer is a 4th layer from the top.
- The main role of the transport layer is to provide the communication services directly to the application processes running on different hosts.
- The transport layer provides a logical communication between application processes running on different hosts. Although the application processes on different hosts are not physically connected, application processes use the logical communication provided by the transport layer to send the messages to each other.
- The transport layer protocols are implemented in the end systems but not in the network routers.

- A computer network provides more than one protocol to the network applications. For example, TCP and UDP are two transport layer protocols that provide a different set of services to the network layer.
- All transport layer protocols provide multiplexing/demultiplexing service. It also provides other services such as reliable data transfer, bandwidth guarantees, and delay guarantees.
- Each of the applications in the application layer has the ability to send a message by using TCP or UDP. The application communicates by using either of these two protocols. Both TCP and UDP will then communicate with the internet protocol in the internet layer. The applications can read and write to the transport layer. Therefore, we can say that communication is a two-way process.



The services provided by the transport layer protocols can be divided into five categories:

- End-to-end delivery
- Addressing
- Reliable delivery
- Flow control
- Multiplexing



End-to-end delivery:

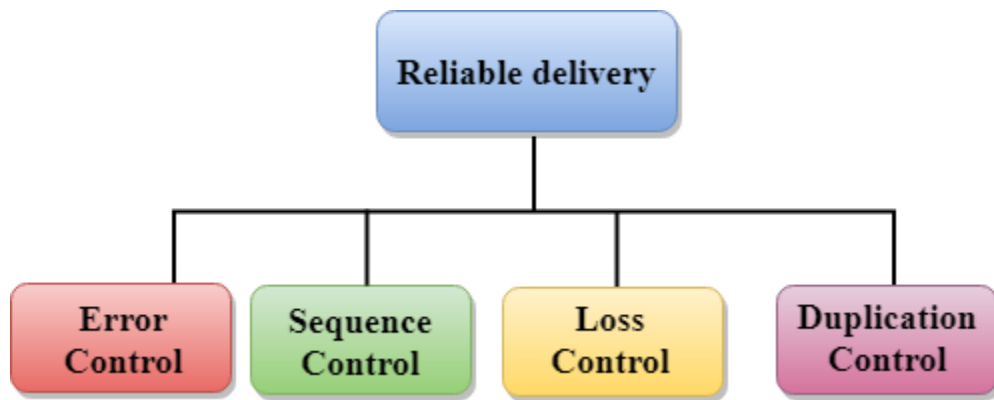
The transport layer transmits the entire message to the destination. Therefore, it ensures the end-to-end delivery of an entire message from a source to the destination.

Reliable delivery:

The transport layer provides reliability services by retransmitting the lost and damaged packets.

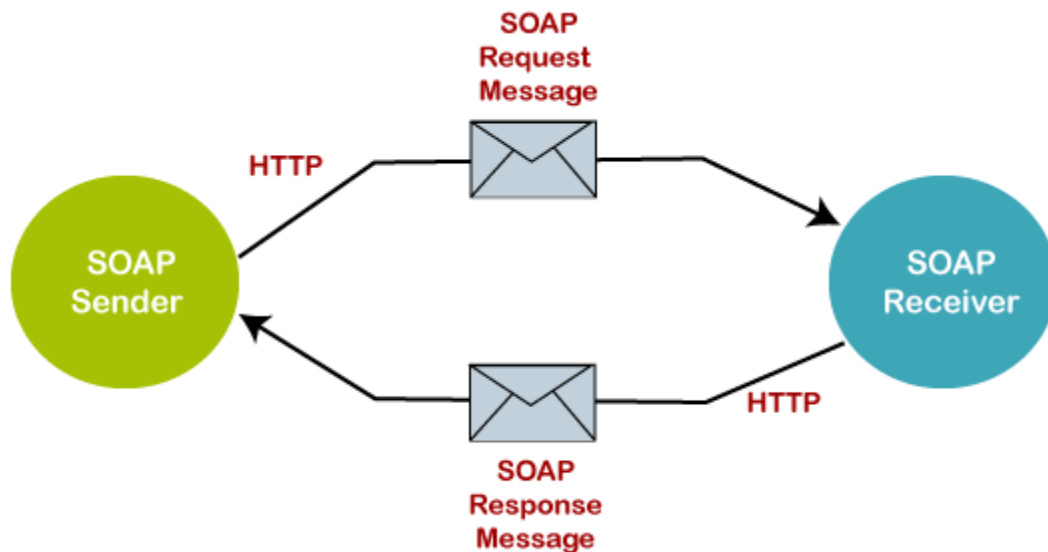
The reliable delivery has four aspects:

- Error control
- Sequence control
- Loss control
- Duplication control



SOAP Web Services

SOAP stands for **Simple Object Access Protocol** is a network platform used in a web service to exchange or communicate data between two different machines on a network. It uses the XML format of data to transfer messages over the [HTTP](#) protocol. In Web services, SOAP allows the user request to interact with other programming languages. In this way, it provides a way to communicate between applications running on different platforms (Operating system), with programming languages and technologies used in web service.



SOAP Message:

The SOAP message contains the following information in the [XML](#) format, as given below.

- It contains information about the message structure and instructions during processing on the network.
- The SOAP contains an envelope that represents the starting and end of the message in the XML format.
- In the message, the header is an optional element that contains application-specific information such as authentication, authorization, and payment etc.
- A fault element is an optional element that shows an error message during the processing of the information.

Characteristics of SOAP

- It is an open standard protocol used in the web service to communicate via internet.
- It is used to broadcast a message over the network.
- It is used to call remote procedures and exchange documents.
- It can be used on any platform and can support multi-languages. So, it is a platform and language independent.
- It uses the XML format to send messages over the HTTP protocol.
- The structure of a SOAP message consists of an envelope, header, and body element.

SOAP Building Block

The SOAP building block describes what XML data is sent to the web service and client application. The following diagram represents the SOAP building block.

SOAP Header: It is an optional element in which the header contains the credentials information such as authorization, authentication, etc. is used during the processing of a SOAP message.

SOAP Body: It is an important element of the SOAP message that contains request and response information in the XML format. It defines the actual content of the message to be sent between the client and the webserver.

SOAP Fault: The SOAP Fault element is an optional element used to display an error message encountered during the transmission of a SOAP message. It holds the status of SOAP messages and errors.

The following are the sub-elements of the SOAP Fault element.

Sub Fault Element Description

<faultcode> It is used to identify the fault code in the SOAP message.

<faultstring> It is used to provide the human-readable description of the error.

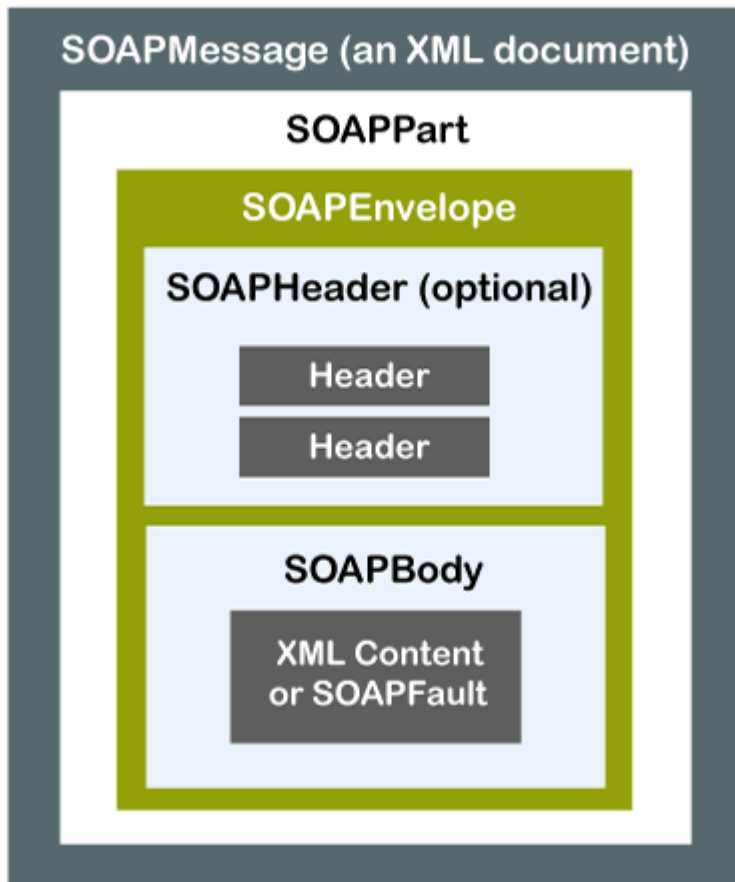
<faultactor> It is an optional element in the SOAP Fault that indicate the fault occurred during processing if the message.

<detail> It is used to hold the application-specific status error of the Body element.

SOAP Structure:

The following code represents the SOAP structure.

```
<?xml version="1.0"?>
<SOAP:Envelope
xmlns:SOAP="http://www.w3.org/2003/05/SOAP-envelope/"
SOAP:encodingStyle="http://www.w3.org/2003/05/SOAP-encoding">
<SOAP:Header> ...
</SOAP:Header>
<SOAP:Body> ...
<SOAP:Fault> ...
</SOAP:Fault>
</SOAP:Body>
</SOAP:Envelope>
```

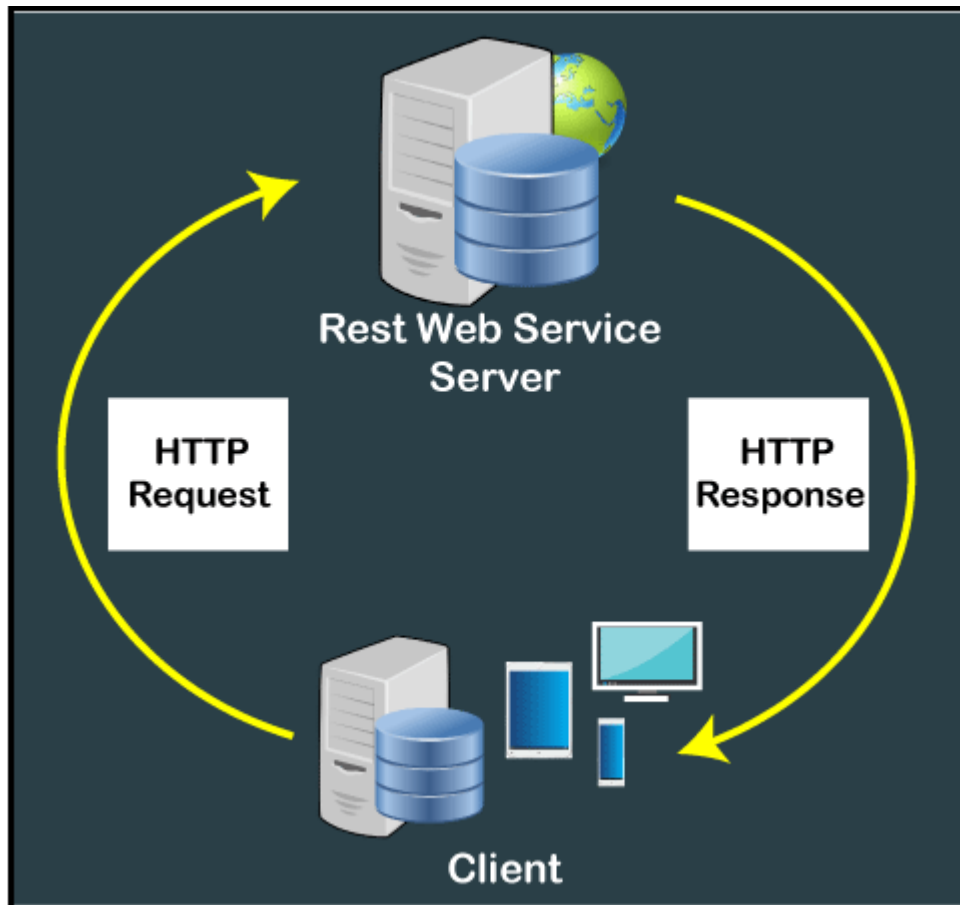


SOAP Envelope: Envelope is used to define the start and end of the SOAP message. It contains the details of the SOAP message. It is an important element of the XML documents.

REST Web Service

REST stands for REpresentational State Transfer. It is an architectural style that provides standards between computer systems on a web service. It is a set of constraints used to create a lightweight, scalable, and maintainable web service that facilitates easy communication. It is a set of constraints used to create a lightweight, scalable and maintainable web service that makes easy communication with each other. All web services are based on the REST; hence it is called a RESTful service. The purpose of developing a [RESTful web service](#) is to make the web service more effective. The Roy Fielding develops it in 2000, who also developed the HTTP protocol. It does not depend on a specific protocol to use, but it's used HTTP/HTTPS.

- It is used to accommodate extra layers like proxy servers, firewalls, gateways, and caching system.
- Code on Demand: It is an optional code on-demand constraints that allow the client to request and execute the server's code. Furthermore, the server can deploy new features to the clients' machine.



REST Characteristics

- It is a Stateless server.
- It supports JSON and XML
- It is simple than SOAP.
- It has well-maintained documentation that reflect each change in the architectures of the REST.
- It is a simple approach to build the client and server service.
- It follows standard protocol such as HTTP, HTTPS and FTP.
- It provides a way to connect with server-side applications.
- It is based on architectural style for designing simple, lightweight, and distributed web services compared to SOAP.

- Examples of REST Systems are -Google Glass API, Amazon web services, Atom, Tesla Model S.

Constraint and the Principles of REST

- Client-Server
- Stateless Server
- Cacheable
- Uniform Interface
- Layered System

Client-Server Model:

- **Client:** Client machines or users send a special request to the webserver and wait for the web server's response.
- **Server:** Server is the collection of web resources that provides different services to multiples clients. It receives multiples requests from the client machines and responds to that request to the client.

Stateless Server: Each request generated from the client-side to the server should contain all the necessary information easily understood by the server.

Cache Response: Any response received from the web server must verify whether the response is cacheable or not. A cache helps both the clients and the servers respond faster to not generate traffic to the web server. For example, when the client repeatedly asks the same request on the server, which is already responded by the server. And again, the same request will go to the server, which increases traffic throughout the network.

Uniform Interface: It is the technique that represents how the REST web service is worked with [HTTP](#) protocol.

- Resource Identifier - URL/URI
- Resource Representation and Manipulation - It defines which the URL identifies data resources. And that data can be manipulated directly from the client URL requested to the Server.
- It helps to manage communication between client and server.

Layered System

HTTP Method of REST web service

GET: It is used to fetch information from the resource server.

POST: It is used to create or insert new information on the resource server.

DELETE: It is used to remove the records or information from the resource server.

Advantage of REST

- REST web services are fast as compared to SOAP because it has no restriction like SOAP. It consumes less bandwidth and resources.
- It is an architectural style for creating lightweight, scalable and maintainable web service.
- It is a language and platform-independent web service that can be written in any programming language and run on Windows, Linux and Mac.
- REST web service helps the client machine access different formats of data such as HTML, JSON, XML, etc. from the webserver.

Difference Between SOAP and REST Web Service

SOAP Web Service	REST Web Service
It stands for Simple Object Access Protocol.	It stands for REpresentational State Transfer.
It is XML based messaging protocol.	It is not a protocol. It is an architectural style for distributed hypermedia system.
It needs more bandwidth and resources for better web performance.	REST requires less bandwidth and resources as compared to SOAP.
SOAP enforces XML as a message format.	It is not specifically applied that the message format must be XML or

	JSON, etc.
It has not great performance as compared to REST.	It has better performance as compared to SOAP, less CPU intensive, lesser code, etc.
SOAP defines its security.	REST inherits security measures from the underlying transport.
It does not support error handling.	It has built-in error handling.
SOAP is a heavyweight XML protocol that requires more coding to send a message.	It is a lightweight, scalable and maintainable.
It cannot be cached.	It can be cached.
SOAP messages are wrapped in an envelope and sent to any transport mechanism such as SMTP, FTP, HTTP or any protocol.	It relies on the HTTP protocol for communication between two machines.

WSDL stands for Web Services Description Language. It is the standard format for describing a web service. WSDL was developed jointly by Microsoft and IBM.

Features of WSDL

- WSDL is an XML-based protocol for information exchange in decentralized and distributed environments.
- WSDL definitions describe how to access a web service and what operations it will perform.
- WSDL is a language for describing how to interface with XML-based services.
- WSDL is an integral part of Universal Description, Discovery, and Integration (UDDI), an XML-based worldwide business registry.
- WSDL is the language that UDDI uses.
- WSDL is pronounced as 'wiz-dull' and spelled out as 'W-S-D-L'.

WSDL Usage

WSDL is often used in combination with SOAP and XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what functions are available on the server. Any special datatypes used are embedded in the WSDL file

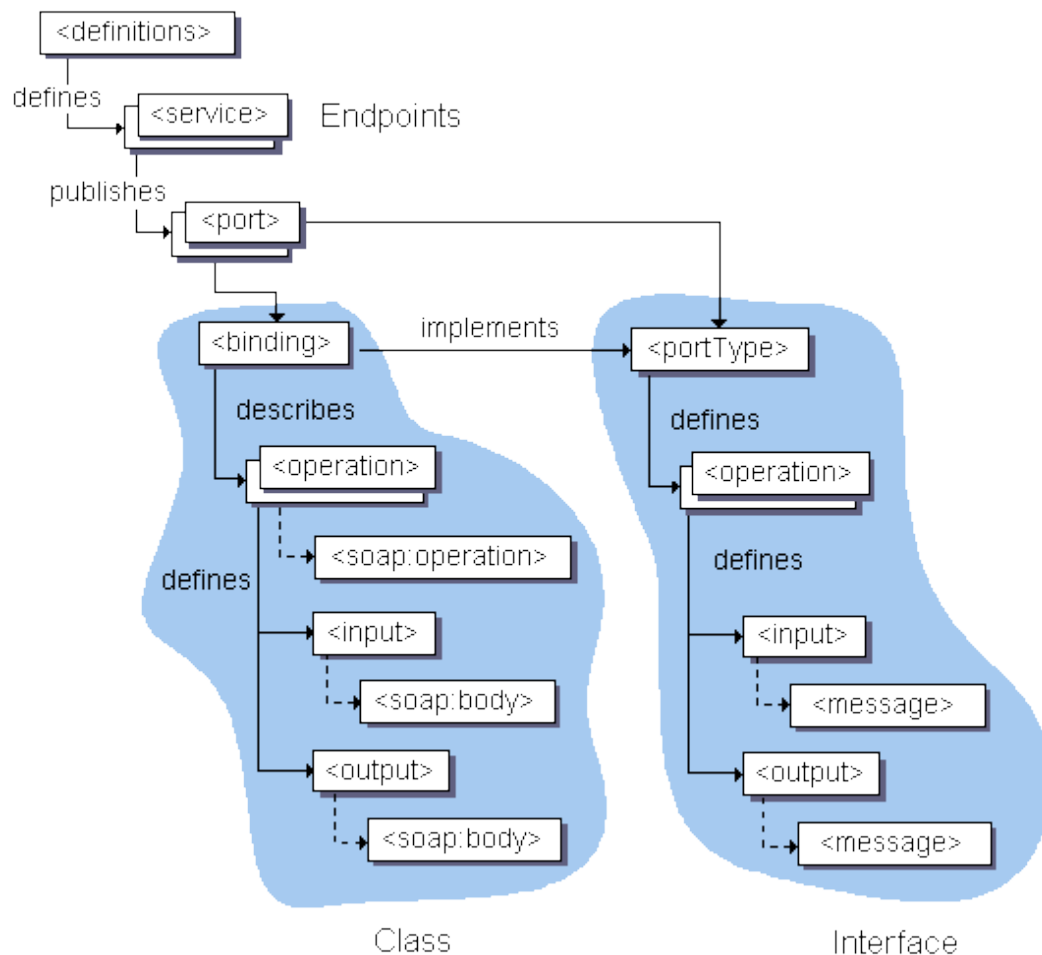
in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL.

History of WSDL

WSDL 1.1 was submitted as a W3C Note by Ariba, IBM, and Microsoft for describing services for the W3C XML Activity on XML Protocols in March 2001.

WSDL 1.1 has not been endorsed by the World Wide Web Consortium (W3C), however it has just released a draft for version 2.0 that will be a recommendation (an official standard), and thus endorsed by the W3C.

Web Services Description Language (WSDL) is an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages. The diagram below illustrates the elements that are present in a WSDL document, and indicates their relationships. To see an example of how this is implemented in a WSDL document .



A WSDL document has a definitions element that contains the other five elements, types, message, portType, binding and service. The following sections describe the features of the generated client code.

WSDL supports the XML Schemas specification (XSD) as its type system.

definitions

Contains the definition of one or more services. JDeveloper generates the following attribute declarations for this section:

- name is optional.
- targetNamespace is the logical namespace for information about this service. WSDL documents can import other WSDL documents, and setting targetNamespace to a unique value ensures that the namespaces do not clash.
- xmlns is the default namespace of the WSDL document, and it is set to <http://schemas.xmlsoap.org/wsdl/>.
- All the WSDL elements, such as <definitions>, <types> and <message> reside in this namespace.
- xmlns:xsd and xmlns:soap are standard namespace definitions that are used for specifying SOAP-specific information as well as data types.
- xmlns:tns stands for this namespace.
- xmlns:ns1 is set to the value of the schema targetNamespace, in the <types> section.
- **types**

Provides information about any complex data types used in the WSDL document. When simple types are used the document does not need to have a types section.

message

An abstract definition of the data being communicated. In the example, the message contains just one part, response, which is of type string, where string is defined by the XML Schema.

operation

An abstract description of the action supported by the service.

portType

An abstract set of operations supported by one or more endpoints.

binding

Describes how the operation is invoked by specifying concrete protocol and data format specifications for the operations and messages.

port

Specifies a single endpoint as an address for the binding, thus defining a single communication endpoint.

service

Specifies the port address(es) of the binding. The service is a collection of network endpoints or ports.