**Web Services**

Web services are components of web application.

Web services can be published, found, and used on the Web such as WSDL, SOAP, UDDI, and RDF.

Web Services

* Web services can be discovered using UDDI
* HTTP and XML is the basis for Web services
* Web services are self-contained and self-describing
* Web services communicate using open protocols
* Web services can be used by other applications

Interoperability has Highest Priority

Web-applications are simply applications that run on the web. These are built around all major standard [the Web browser](http://www.w3schools.com/Webservices/ws_intro.asp) and can be used by any browser on any platform. For these platforms to work together, Web applications were developed.

When all major platforms could access the Web using Web browsers, different platforms probably couldn't interact.

Web Services take Web applications to the Next Level. Web application can publish its function or message to the rest of the world.

Web services use XML to code and to decode data, and SOAP to transport it (using open protocols).

With Web services, your [accounting](http://www.w3schools.com/Webservices/ws_intro.asp) department's Win 2k server's billing system can connect with your IT supplier's UNIX server.

Web Services have Two Types of Uses

**Reusable application-components.**

There are things applications need very often.

Web services can offer reusable application-components like: currency conversion, weather reports, or even language translation as services.

**Connect existing applications**

Web services can help to be interoperable by giving different applications a way to link their data.

With Web services you can exchange data between different applications and different platforms.

Any application can have a Web Service component.

Web Services can be created regardless of programming language.

[**http://www.w3schools.com/webservices/ws\_intro.asp**](http://www.w3schools.com/webservices/ws_intro.asp)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

WSDL

WSDL stands for Web Service Description Language for describing web services and how to access them.

WSDL is XML-based Web service document specified the location of the service and the operations (or methods) the service exposes.

WSDL Documents

A WSDL document is just a simple XML document.

It contains set of definitions to describe a web service.

The WSDL Document Structure

A WSDL document describes a set of definition of web service using these major elements:

|  |  |
| --- | --- |
| Element | Description |
| <types> | A container for data type definitions used by the web service |
| <message> | A typed definition of the input and output data being communicated |
| <portType> | A set of operations supported by one or more endpoints |
| <binding> | A protocol and data format specification for a particular port type |
|  |  |

The main structure of a WSDL document is:

<definitions><types>  
  data type definitions..   
</types>  
<message>  
  definition of the data being communicated…   
</message><portType>  
  set of operations…  
</portType**>**<binding>  
  protocol and data format specification…  
</binding></definitions>

A WSDL document can also contain other elements, like extension elements, and a service element that makes it possible to group together the definitions of several **web services** in one single WSDL document.

WSDL Types

The **<types>** element defines the data types that are used by the web service.

For maximum platform neutrality, WSDL uses XML Schema syntax to **define data** types.

WSDL Messages

The **<message>** element defines the data elements of an operation.

Each message can consist of one or more parts. The parts can be compared to the parameters of a function call in a traditional [programming](http://www.w3schools.com/Webservices/ws_wsdl_documents.asp) language.

WSDL Ports

The **<portType>** element is the most important WSDL element.

It describes a web service, the operations that can be performed, and the messages that are involved.

The <portType> element can be compared to a functional library (or a module, or a class) in a traditional [**programming**](http://www.w3schools.com/Webservices/ws_wsdl_documents.asp) **language**.

WSDL Bindings

The **<binding>** element defines the data format and protocol for each port type.

WSDL Example

<message name="getTermRequest">  
  <part name="term" type="xs:string"/>  
</message>  
  
<message name="getTermResponse">  
  <part name="value" type="xs:string"/>  
</message>  
  
<portType name="glossaryTerms">  
  <operation name="getTerm">  
    <input message="getTermRequest"/>  
    <output message="getTermResponse"/>  
  </operation>  
</portType>

In this example the **<portType>** element defines "glossaryTerms" as the name of a **port**, and "getTerm" as the name of an **operation**.

The "getTerm" operation has an **input message** called "getTermRequest" and an **output message** called "getTermResponse".

The **<message>** elements define the **parts** of each message and the associated data types.

Compared to traditional programming, glossaryTerms is a function library, "getTerm" is a function with "getTermRequest" as the input parameter, and getTermResponse as the return parameter.

WSDL PortType

The **<portType>** element is the most important WSDL element.

WSDL - The <portType> Element

The <portType> element defines **a web service**, the **operations** that can be performed, and the **messages** that are involved.

<portType> defines the **connection point** to a web service. It can be compared to a function library (or a module, or a class) in a traditional [**programming**](http://www.w3schools.com/Webservices/ws_wsdl_ports.asp) **language**. Each operation can be compared to a function in a traditional [programming](http://www.w3schools.com/Webservices/ws_wsdl_ports.asp) language.

Operation Types

The request-response type is the most common operation type, but WSDL defines four types:

|  |  |
| --- | --- |
| **Type** | **Definition** |
| One-way | The operation can receive a message but will not return a response |
| Request-response | The operation can receive a request and will return a response |
| Solicit-response | The operation can send a request and will wait for a response |
| Notification | The operation can send a message but will not wait for a response |

One-Way Operation

A one-way operation example:

<message name="newTermValues">  
  <part name="term" type="xs:string"/>  
  <part name="value" type="xs:string"/>  
</message>  
  
<portType name="glossaryTerms">  
  <operation name="setTerm">  
    <input name="newTerm" message="newTermValues"/>  
  </operation>  
</portType >

In the example above, the portType "glossaryTerms" defines a one-way operation called "setTerm".

The "setTerm" operation allows input of new glossary **terms** messages using a "newTermValues" message with the input parameters "term" and "value". However, no output is defined for the operation.

Request-Response Operation

A request-response operation example:

<message name="getTermRequest">  
  <part name="term" type="xs:string"/>  
</message>  
  
<message name="getTermResponse">  
  <part name="value" type="xs:string"/>  
</message>  
  
<portType name="glossaryTerms">  
  <operation name="getTerm">  
    <input message="getTermRequest"/>  
    <**output** message="getTermResponse"/>  
  </operation>  
</portType>

In the example above, the portType "glossaryTerms" defines a request-response operation called "getTerm".

The "getTerm" operation requires an input message called "getTermRequest" with a **parameter** called "term", and will return an output message called "getTermResponse" with a parameter called "value".

WSDL Bindings

WSDL **bindings** defines the message format and protocol details for a **web service**.

Binding to SOAP

A request-response operation example:

<message name="getTermRequest">  
  <part name="term" type="xs:string"/>  
</message>  
  
<message name="getTermResponse">  
  <part name="value" type="xs:string"/>  
</message>  
  
<portType name="glossaryTerms">  
  <operation name="getTerm">  
    <input message="getTermRequest"/>  
    <output message="getTermResponse"/>  
  </operation>  
</portType>  
  
<binding type="glossaryTerms" name="b1">  
   <**soap**:binding style="document"  
   **transport**="http://schemas.xmlsoap.org/soap/http" />  
   <operation>  
     <soap:operation soapAction="http://example.com/getTerm"/>  
     <input><soap:body use="literal"/></input>  
     <output><soap:body use="literal"/></output>  
  </operation>  
</binding>

The **binding** element has two attributes - name and type.

The name attribute (you can use any name you want) defines the name of the binding, and the type attribute points to the port for the binding, in this case the "glossaryTerms" port.

The **soap:binding** element has two attributes - style and transport.

The style attribute can be "rpc" or "document". In this case we use document. The transport attribute defines the SOAP protocol to use. In this case we use HTTP.

The **operation** element defines each operation that the portType exposes.

For each operation the corresponding SOAP action has to be defined. You must also specify how the **input and output** are encoded. In this case we use "literal".

WSDL and UDDI

[Universal](http://www.w3schools.com/Webservices/ws_wsdl_uddi.asp) Description, Discovery and Integration (UDDI) is a directory service where businesses can [register](http://www.w3schools.com/Webservices/ws_wsdl_uddi.asp) and search for Web services.

UDDI

UDDI is a platform-independent framework for describing services, [discovering businesses](http://www.w3schools.com/Webservices/ws_wsdl_uddi.asp), and integrating **business services** by using the **Internet**.

* UDDI is a directory for storing information about web services
* UDDI is a directory of web service interfaces described by WSDL
* UDDI communicates via **SOAP**
* UDDI is built into the [**Microsoft**](http://www.w3schools.com/Webservices/ws_wsdl_uddi.asp) .NET platform

What is UDDI Based On?

UDDI uses World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF) Internet standards such as XML, HTTP, and DNS protocols.

UDDI uses WSDL to describe interfaces to web services

Additionally, cross platform [programming](http://www.w3schools.com/Webservices/ws_wsdl_uddi.asp) features are addressed by adopting SOAP, known as XML Protocol messaging specifications found at the W3C Web site.

UDDI Benefits

Any industry or businesses of all sizes can benefit from UDDI.

Before UDDI, there was no Internet standard for businesses to reach their customers and partners with information about their products and services. Nor was there a method of how to integrate into each other's systems and processes.

Problems the UDDI specification can help to solve:

* Making it possible to discover the right business from the millions currently online
* Defining how to enable commerce once the preferred business is discovered
* Reaching new customers and increasing access to current customers
* Expanding offerings and extending market reach
* Solving customer-driven need to remove barriers to allow for rapid participation in the global Internet economy
* Describing services and business processes programmatically in a single, open, and [secure](http://www.w3schools.com/Webservices/ws_wsdl_uddi.asp) environment

UDDI is used:

If the industry published an UDDI standard for **flight** rate checking and reservation, **airlines** could register their flight services into an UDDI directory. Travel agencies could then search the UDDI directory to find the airline's reservation interface. When the interface is found, [the travel agency](http://www.w3schools.com/Webservices/ws_wsdl_uddi.asp) can communicate with the service immediately because it uses a well-defined reservation interface.

Who is Supporting UDDI?

UDDI is a cross-industry effort driven by all major platform and [**software**](http://www.w3schools.com/Webservices/ws_wsdl_uddi.asp) providers like Dell, [Fujitsu](http://www.w3schools.com/Webservices/ws_wsdl_uddi.asp), HP, Hitachi, IBM, Intel, Microsoft, Oracle, SAP, and Sun, as well as a large community of marketplace operators, and e-business leaders.

Over 220 companies are members of the UDDI community.

[**http://www.w3schools.com/Webservices/ws\_wsdl\_intro.asp**](http://www.w3schools.com/Webservices/ws_wsdl_intro.asp)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**