

Z-blog

O.R.Manoj Niranthaka. IT17048088

FRIDAY, MARCH 15, 2019

Web Services and SOAP

Compare and contrast web applications with web services, indicating the need for web services

Web applications

are defined by being interactive. You're supposed to use a web application in order to perform a function and use some of the web applications features. Lots of data exactly. People are just supposed to use them in order to perform additional tasks, using their features to accomplish something. You use a web application to play a game.

The browser capabilities involved with web applications are significantly more high-tech, which is one reason why it's usually harder for people to design a web application that handles more data, and web applications are all about doing things. One of your actions is probably going to be getting more information or learning more information from the information from a website.

The user interface of a web application is also usually much more complicated than the user interface of a website. Websites might have tags and categories that you can click on and learn any potentially complicated tasks in order to use websites. Web applications often require step-by-step guides, or you're not going to be able to complete a task without a guide.

The setup of websites are completely different from web applications in most cases. The rhythm of typing in the address, loading websites, and going back and forth between websites is different from web applications.

Web applications, unsurprisingly, are usually harder to design and create than websites. Lots of people have their own websites today, and this was the case even before web applications can more easily make money off of them because it takes more work to learn how to code and create a web application.

Web service

Web service is a technology by which two or more remote web applications interact with each other over network/internet. It can be implemented using Java, JavaScript, etc. Web services allow programs to communicate and collaborate with each other.

A web service is essentially a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written for 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. The popularity of web services (both Windows and Linux applications) is due to the use of open standards (XML, SOAP, HTTP).

All the standard Web Services works using following components:

SOAP (Simple Object Access Protocol)

UDDI (Universal Description, Discovery and Integration)

WSDL (Web Services Description Language)

It works somewhat like this:

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 up 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.

Discuss what WSDL is and the use of it in the context of web services

Web Services Description Language (WSDL) is a format for describing a Web Services interface. It is a way to describe services and how they should be bound to a particular protocol.

***Definitions**

***Operations**

***Service bindings**

Definitions are generally expressed in XML and include both data type definitions and message definitions that use the data type definitions. These definitions This agreement could be within an organization or between organizations. Vocabularies within an organization could be designed specifically for that organization's vocabulary. If data type and message definitions need to be used between organizations, then most likely an industry-wide vocabulary will be used. For more information, see the WSDL specification.

XML, however, is not necessarily required for definitions. The OMG Interface Definition Language (IDL), for example, could be used instead of XML. If a different format would need to agree on the format as well as the vocabulary. Nevertheless, over time, XML-based vocabularies and messages are likely to dominate. XML Namespaces are used in the definitions, operations, and service bindings.

Operations describe actions for the messages supported by a Web service. There are four types of operations:

One-way: Messages sent without a reply required

Request/response: The sender sends a message and the receiver sends a reply.

Solicit response: A request for a response. (The specific definition for this action is pending.)

Notification: Messages sent to multiple receivers. (The specific definition for this action is pending.)

Operations are grouped into port types. Port types define a set of operations supported by the Web service.

Service bindings connect port types to a port. A port is defined by associating a network address with a port type. A collection of ports defines a service. This information can be used. These other forms could include CORBA Internet Inter-ORB Protocol (IIOP), DCOM, .NET, Java Message Service (JMS), or WebSphere MQ to name a few.

Explain the fundamental properties of a WSDL document and the use of WSDL document in web service

-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.

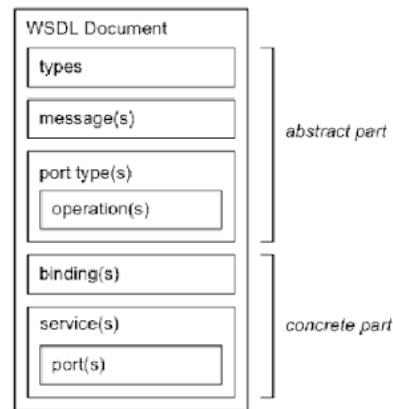
-WSDL is the language that UDDI uses.

-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 use the information in the WSDL file to determine the service's capabilities. 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 service's operations.

Discuss the structure of the WSDL document, explaining the elements in WSDL



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

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 is required to avoid a namespace 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 in the types section.

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.

The main structure of a WSDL document looks like this

```

<definitions>
  <types>
    definition of types.....
  </types>

  <message>
    definition of a message....
  </message>

  <portType>
    <operation>
      definition of a operation.....
    </operation>
  </portType>

  <binding>
    definition of a binding....
  </binding>

  <service>
    definition of a service....
  </service>
</definitions>

```

Compare the PortType and operation elements in WSDL with the java equivalences

portType

The <portType> element combines multiple message elements to form a complete one-way or round-trip operation.

For example, a <portType> can combine one request and one response message into a single request/response operation. This is most commonly used in SOAF

```

<portType name = "Hello_PortType">
  <operation name = "sayHello">
    <input message = "tns:SayHelloRequest"/>
    <output message = "tns:SayHelloResponse"/>
  </operation>
</portType>

```

operation

element provides binding information from the abstract operation to the concrete SOAP operation.

The following example illustrates the use of the SOAP operation element.

```

<definitions ....>
  <binding ....>
    <operation ....>
      <soap12:operation soapAction="xs:anyURI" ?
        soapActionRequired="xs:boolean" ?
        style="rpc|document" ?
        wsdl:required="xs:boolean" ? /> ?
    </soap12:operation>
  </binding>
</definitions>

```

Compare and contrast the binding and service elements in WSDL

binding

The <binding> element provides specific details on how a portType operation will actually be transmitted over the wire.

- *The bindings can be made available via multiple transports including HTTP GET, HTTP POST, or SOAP.
- *The bindings provide concrete information on what protocol is being used to transfer portType operations.
- *The bindings provide information where the service is located.
- *For SOAP protocol, the binding is <soap:binding>, and the transport is SOAP messages on top of HTTP protocol.
- *You can specify multiple bindings for a single portType.

The binding element has two attributes : name and type attribute.

```
<binding name = "Hello_Binding" type = "tns:Hello_PortType">
```

The name attribute defines the name of the binding, and the type attribute points to the port for the binding, in this case the "tns:Hello_PortType" port.

service

The <service> element defines the ports supported by the web service. For each of the supported protocols, there is one port element. The service element is a

-Web service clients can learn the following from the service element –

- *where to access the service,
- *through which port to access the web service, and
- how the communication messages are defined.

-The service element includes a documentation element to provide human-readable documentation.

Here is a piece of code from the Example

```
<service name = "Hello_Service">
  <documentation>WSDL File for HelloService</documentation>
  <port binding = "tns:Hello_Binding" name = "Hello_Port">
    <soap:address
      location = "http://www.examples.com/SayHello/">
  </port>
```

Explain how SOAP is used with HTTP

A binding of SOAP to a transport protocol is a description of how a SOAP message is to be sent using that transport protocol

the typical binding for SOAP is HTTP

SOAP can use GET or POST ,with GET ,the request is not a SOAP message but the response is a SOAP message,with POST both request and responses are SOAP messages

SOAP uses the same error and status codes as those used in HTTP so that HTTP responses can be directly interpreted by a SOAP module

Explain the structure of SOAP message in message oriented communication, indicating the elements used

A SOAP message is an ordinary XML document containing the following elements

Envelope – Defines the start and the end of the message. It is a mandatory element.

Header – Contains any optional attributes of the message used in processing the message, either at an intermediary point or at the ultimate end-point. It is an optional element.

Body – Contains the XML data comprising the message being sent. It is a mandatory element.

Fault – An optional Fault element that provides information about errors that occur while processing the message.

SOAP Message Structure

```
<?xml version = "1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV = "http://www.w3.org/2001/12/soap-envelope"
```

SOAP-ENV:encodingStyle = "<http://www.w3.org/2001/12/soap-encoding>">

<SOAP-ENV:Header>

...

...

</SOAP-ENV:Header>

<SOAP-ENV:Body>

...

...

<SOAP-ENV:Fault>

...

...

</SOAP-ENV:Fault>

...

</SOAP-ENV:Body>

</SOAP_ENV:Envelope>

Discuss the importance of the SOAP attachments, explaining the MIME header

- SOAP messages may have one or more attachments
- each AttachmentPart object has a MIME header to indicate the type of data it contains.
- it may also have additional MIME headers to identify it or to give its location, which can be useful when there are multiple attachments
- when a SOAP message has one or more AttachmentPart objects, its SOAPPart object may or may not contain message content

Identify different set of frameworks/libraries for SOAP web service development, in different environments (Java, .Net, PHP, etc...)

| | |
|-----------------------------------|------------|
| Apache Axis | - java/c++ |
| Java Web Service Development Pack | - java |
| .NET Framework | - c# |
| WSO2 WSF/PHP | - PHP |
| Zend Framework | -PHP |

Explain the annotations in JAX-WS, providing examples of their use

Java™ API for XML-Based Web Services (JAX-WS) relies on the use of annotations to specify metadata associated with web services implementations and to how a server-side service implementation is accessed as a web service or how a client-side Java class accesses web services.

examples: @WebService

@SOAPBinding

@WebMethod

@WebResult

@WebServiceClient


Posted by **ManojNiranthaka** at **10:43 AM**



No comments:

Post a Comment

Enter your comment...

Comment as: ManojNiranthal ▾

Publish

Preview

Home

Subscribe to: [Post Comments \(Atom\)](#)

ABOUT ME



ManojNiranthaka
[View my complete profile](#)

BLOG ARCHIVE

- ▼ 2019 (5)
 - ▼ March (2)
 - Web Services and SOAP
 - Distributed systems
 - February (3)



Picture Window theme. Theme images by richcano. Powered by **Blogger**.