Module -3 WSDL

- What is WSDL?
- Why WSDL?
- Structure of WSDL
- WSDL Elements
  - Definition
  - Target Namespace
  - Data type
  - Message
  - Part type
  - Bindings
  - Service
- Demonstration on elements of WSDL

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'.

What is WSDL

- Web Services Description Language (WSDL) is an XML-based file that basically tells the client application what the web service does.

- The WSDL file is used to describe in a nutshell what the web service does and gives the client all the information required to connect to the web service and use all the functionality provided by the web service.

One key thing to note here is that definition of messages, which is what is passed by the SOAP protocol is actually defined in the WSDL document.

The WSDL document actually tells a client application what are the types of SOAP messages which are sent and accepted by the Web service.

In other words, the WSDL is just like a postcard which has the address of a particular location. The address provides the details of the person who delivered the postcard. Hence, in the same way, the WSDL file is the postcard, which has the address of the web service which can deliver all the functionality that the client wants.

```
<!-- WSDL definition structure -->
<definitions
                name="Guru99Service"
        targetNamespace=http://example.org/math/
        xmlns=http://schemas.xmlsoap.org/wsdl/>
        <!-- abstract definitions -->
                <types> ...
                        <message> ...
                        <portType> ...

   <!-- concrete definitions -->
                <binding> ...
                <service> ...
</definition>
```

The WSDL file contains the following main parts

1. The **<types>** tag is used to define all the complex datatypes, which will be used in the message exchanged between the client application and the web service. This is an important aspect of the client application, because if the web service works with a complex data type, then the client application should know how to process the complex data type. Data types such as float, numbers, and strings are all simple data types, but there could be structured data types which may be provided by the web service.

For example, there could be a data type called EmployeeDataType which could have 2 elements called "EmployeeName" of type string and "EmployeeID" of type number or integer. Together they form a data structure which then becomes a complex data type.

2. The **<messages>** tag is used to define the message which is exchanged between the client application and the web server. These messages will explain the input and output operations which can be performed by the web service. An example of a message can be a message which accepts the EmployeeID of an employee, and the output message can be the name of the employee based on the EmpoyeeID provided.

3. The **<portType>** tag is used to encapsulate every input and output message into one logical operation. So there could be an operation called "GetEmployee" which combines the input message of accepting the EmployeeID from a client application and then sending the EmployeeName as the output message.

4.The **<binding>** tag is used to bind the operation to the particular port type. This is so that when the client application calls the relevant port type, it will then be able to access the operations which are bound to this port type. Port types are just like interfaces. So if a client application needs to use a web service they need to use the binding information to ensure that they can connect to the interface provided by that web service.

5.The **<service>** tag is a name given to the web service itself. Initially, when a client application makes a call to the web service, it will do by calling the name of the web service. For example, a web service can be located at an address such as **http://localhost/Guru99/Tutorial.asmx** . The service tag will actually have the URL defined as **http://localhost/Guru99/Tutorial.asmx**, which will actually tell the client application that there is a web service available at this location.

The <definitions> element must be the root element of all WSDL documents. It defines the name of the web service.

Here is the piece of code from the last chapter that uses the definitions element.

```
<definitions name="HelloService"
targetNamespace="http://www.examples.com/wsdl/HelloService.wsdl"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://www.examples.com/wsdl/HelloService.wsdl"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
................................................ </definitions>
```

From the above example, we can conclude that definitions −

Is a container of all the other elements.

Specifies that this document is called HelloService.

Specifies a targetNamespace attribute. The targetNamespace is a convention of XML Schema that enables the WSDL document to refer to itself. In this example, we have specified a targetNamespace of http://www.examples.com/wsdl/HelloService.wsd.

Why WSDL

A web service is an important component in building modern day web applications. Their main purpose is to allow multiple applications built on various programming languages to talk to each other. For instance, we can have a .Net web application talks to a Java application via a Web service.

A web service has the following key features

- It is built using the XML programming language. Almost all modern day technologies such as .Net and Java have corresponding commands that have the ability to work with XML. Hence, XML was taken as the most appropriate language for building web services.
- Web services communicate over HTTP. HTTP is a protocol used by all web-based applications. Hence, it just made sense to ensure that Web services also had the ability to work over the HTTP protocol.
- Web services conform to a particular language specification. This specification is set by the W3C, which is the governing body for all web standards.
- Web services have a description language known as WSDL, which is used to describe the web service.

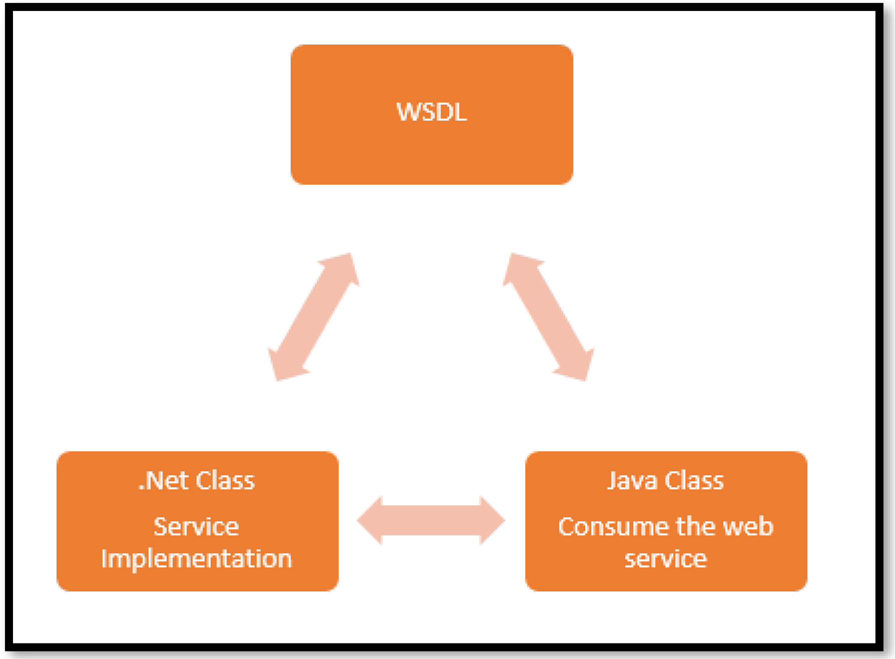
The WSDL file is written in plain old XML. The reason that it is in XML is so that the file can be read by any programming language.
So if the client application was written in .Net, it would understand the XML file. Similarly, if the client application was written in the Java programming language then also it would be able to interpret the WSDL file.

The WSDL file is what binds everything together. From the above diagram, you can see that you can create a web service in the .Net language.

So this is where the service gets implemented. If you did not have the WSDL file and wanted a Java class to consume the web service, you would need a lot of coding effort to achieve this.

But now with the WSDL file which is in XML, which can be understood by any programming language, you can now easily have a Java class consume the .Net web service.

Hence, the amount of coding effort is greatly reduced.



WSDL

.Net Class
Service Implementation

Java Class
Consume the web service

WSDL Message Part

The WSDL consists of a section called "messages" which is denoted by the **<message>** element.

This element is basically used to describe the data that gets exchanged between the web service and the client application.

Each web service will always have **2 types** of messages, One is for the input of the web service, and the other is for the output of the web service.

The input is used to describe the parameters which are accepted by the web service. This is an important aspect of the client application so that it knows the values to be sent as parameters to the web service.

The other type of message is the output message which tells what results are provided by the web service.

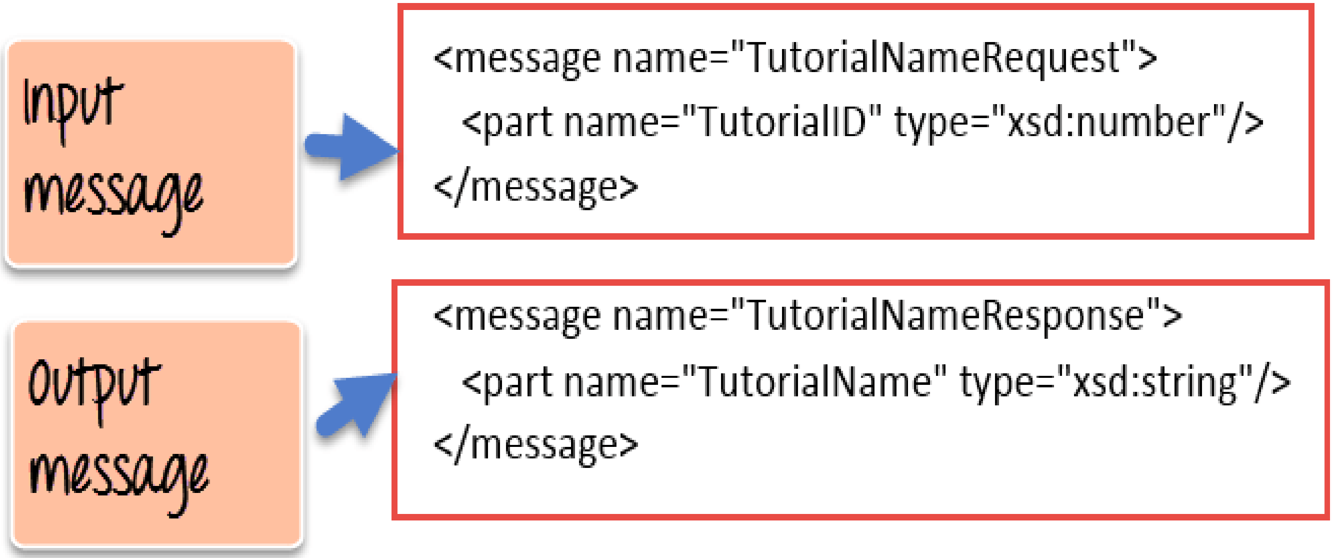
Each message, in turn, will have a **<part>** element which is used to describe the parameter used by the input and output message.

Below is a simple example, of what a message for a web service looks like. The functionality of the web service is to provide the name of a "Tutorial" once a "Tutorial ID" is submitted as a parameter to the web service.

1. As we can see the web service has 2 messages, one for the input and the other for the output.

2. The input message is known as TutorialNameRequest which has one parameter called TutorialID. This parameter is of the type number which is specified by the xsd:number type

3. The output message is known as TutorialNameResponse which has one parameter called TutorialName. This parameter is of the type string which is specified by the xsd:string type



```
<message name="TutorialNameRequest">
  <part name="TutorialID" type="xsd:number"/>
</message>
```

```
<message name="TutorialNameResponse">
  <part name="TutorialName" type="xsd:string"/>
</message>
```

Port Type Binding

Ports are used in WSDL to define one complete operation which is offered by the web service.

In the previous topic, we saw that our web service provided 2 messages, one for the input called "TutorialNameRequest" and the other for the output called "TutorialNameResponse." Together the input and output message form is known as one complete operation.

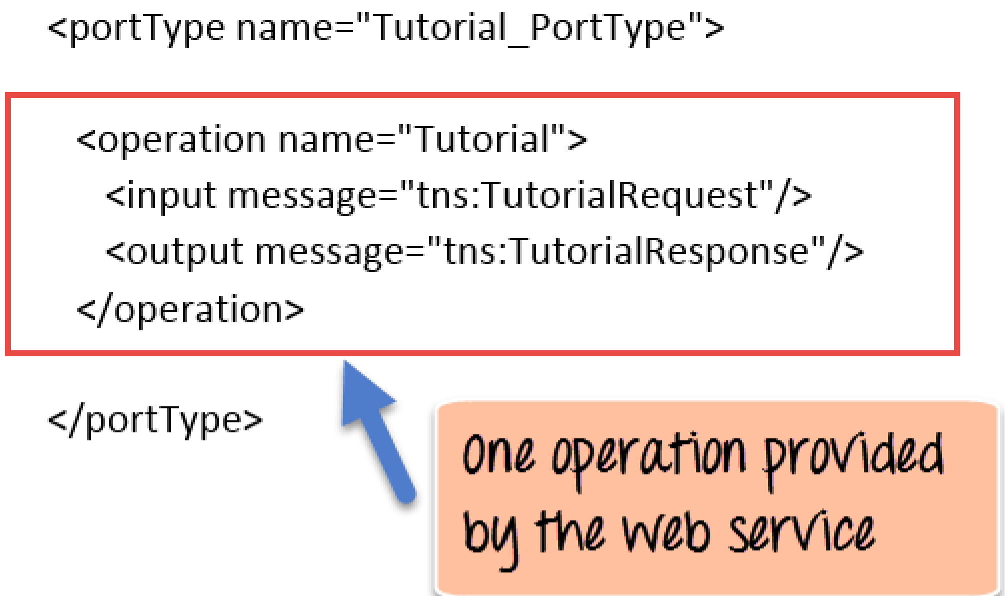
WSDL provides an element called **<portType>** which is used to define the operations provided by the Web service.

So, from the image, we can note the following:

1. The name of the port Type which encapsulates the operation is given as "Tutorial_PortType.”

2. The operation itself is given a name of "Tutorial". So our operation basically provides a TutorialName if the TutorialID is given as an input parameter.

3. Next is our 2 messages, one for the input and the other for the output which forms our operation



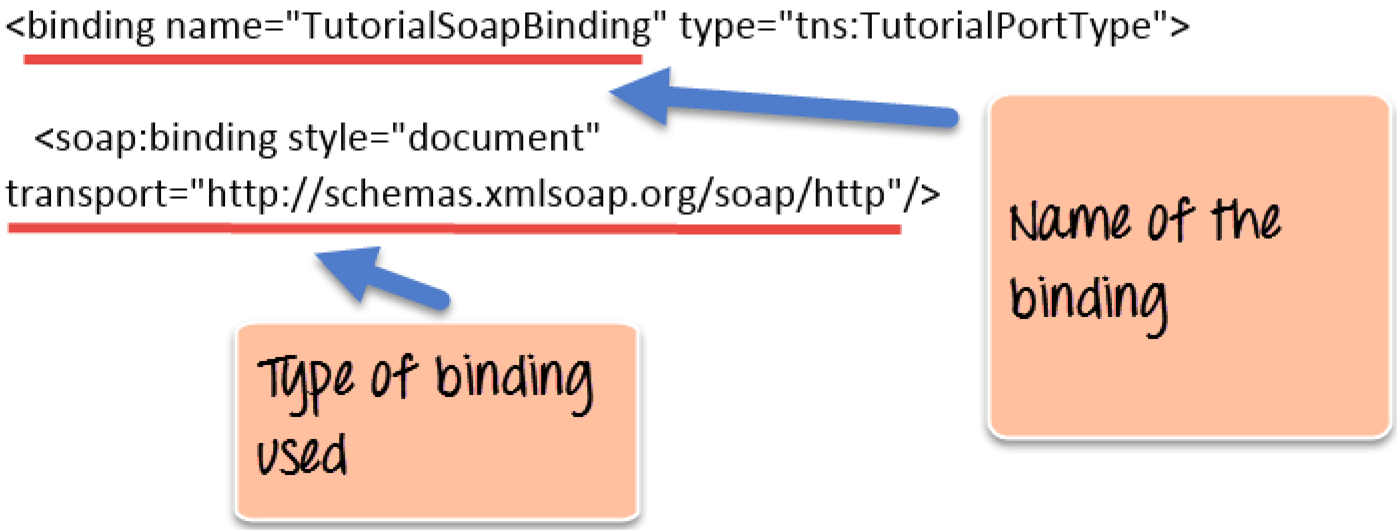
```
<portType name="Tutorial_PortType">

  <operation name="Tutorial">
    <input message="tns:TutorialRequest"/>
    <output message="tns:TutorialResponse"/>
  </operation>

</portType>
```

One operation provided by the web service

In addition to the **<portType>** element, there is also the **<binding>** element which is used to define how the messages will be transferred.

1. The image shows that the binding consists of a binding name which in our case is given as "TutorialSoapBinding". Binding in simple terms is the information which the client application uses to actually bind itself to the web service. Once it is actually bound to the web service, it then has the ability to call the various operations that are exposed by the web service.

2. The transport layer is given as http:// which means that the messages which will transfer over the HTTP protocol.



```
<binding name="TutorialSoapBinding" type="tns:TutorialPortType">

  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
```

Creating WSDL File

The WSDL file gets created whenever a web service is built in any programming language.

Since the WSDL file is pretty complicated to be generated from plain scratch, all editors such as Visual Studio for .Net and Eclipse for Java automatically create the WSDL file.