Title: SOAP Web Services (Simple Object Access Protocol)

What is SOAP (Simple Object Access Protocol)

SOAP is an XML-based protocol for accessing web services over HTTP. It has some specification which could be used across all applications.

SOAP was developed as an intermediate language so that applications built on various programming languages could talk easily to each other and avoid the extreme development effort.

SOAP Introduction

The number of applications which are built on different programming languages for example java, .Net, PHP. Exchanging data between applications is crucial networked world. But data exchange between these heterogeneous applications would be complex. So will be the complexity of the code to accomplish this data exchange.

One of the methods used to combat this complexity is to use XML (Extensible Markup Language) as the intermediate language for exchanging data between applications. Every programming language can understand the XML markup language. Hence, XML was used as the underlying medium for data exchange. But there are no standard specifications on use of XML across all programming languages for data exchange. That is where SOAP comes in.

SOAP was designed to work with XML over HTTP and have some sort of specification which could be used across all applications.

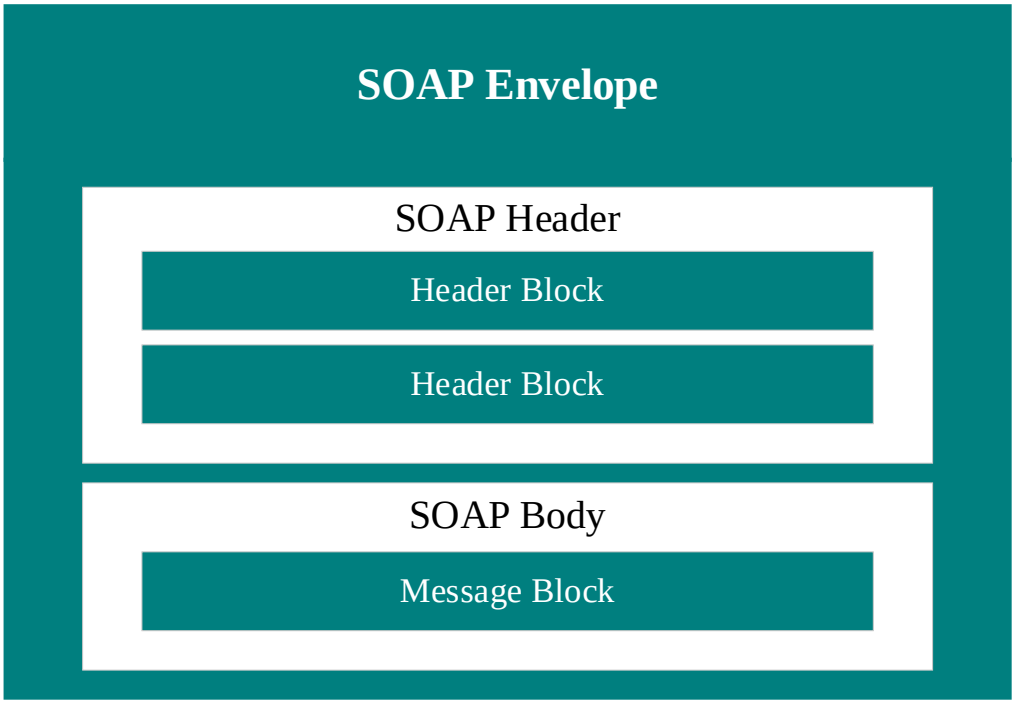
Advantages of SOAP

- When developing Web services, you need to have some of language which can be used for web services to talk with client applications. SOAP is the perfect medium which was developed in order to achieve this purpose. This protocol is also recommended by the W3C consortium which is the governing body for all web standards.
- SOAP is a light-weight protocol that is used for data interchange between applications
- SOAP is designed to be platform independent and is also designed to be operating system independent.
- It works on the HTTP protocol –SOAP works on the HTTP protocol, which is the default protocol used by all web applications.

SOAP Building Blocks

The SOAP specification defines something known as a "SOAP message" which is what is sent to the web service and the client application.

The diagram below shows the various building blocks of a SOAP message.



The SOAP message is nothing but mere an XML document which has the below components:

- An Envelope element that identifies the XML document as a SOAP message – This is the containing part of the SOAP message and is used to encapsulate all the details in the SOAP message. This is the root element in the SOAP message.
- A Header element that contains header information – The header element can contain information such as authentication credentials which can be used by the calling application. It can also contain the definition of complex types which could be used in the SOAP message. By default, the SOAP message can contain parameters which could be of simple types such as strings and numbers, but can also be a complex object type.

  Example of complex type

```
<xsd:complexType>
<xsd:sequence>
<xsd:element name="Tutorial Name" type="string"/>
<xsd:element name="Tutorial Description" type="string"/>
</xsd:sequence>
</xsd:complexType>
```

- A Body element that contains call and response information – This element is what contains the actual data which needs to be sent between the web service and the calling application.

```
<soap:Body>
  <GetTutorialInfo>
        <TutorialName>Web Services</TutorialName>
        <TutorialDescription>All about web services</TutorialDescription>
  </GetTutorialInfo>
</soap:Body>
```

SOAP Message Structure

One thing to note is that SOAP messages are normally auto-generated by the web service when it is called.

Whenever a client application calls a method in the web service, the web service will automatically generate a SOAP message which will have the necessary details of the data which will be sent from the web service to the client application.

1. As seen from the above SOAP message, the first part of the SOAP message is the envelope element which is used to encapsulate the entire SOAP message.
2. The next element is the SOAP body which contains the details of the actual message.
3. Our message contains a web service which has the name of "Guru99WebService".
4. The "Guru99Webservice" accepts a parameter of the type 'int' and has the name of TutorialID.

the above SOAP message will be passed between the web service and the client application.

SOAP Envelope Element

The first bit of the building block is the SOAP Envelope.

The SOAP Envelope is used to encapsulate all of the necessary details of the SOAP messages, which are exchanged between the web service and the client application.

The SOAP envelope element is used to indicate the beginning and end of a SOAP message. This enables the client application which calls the web service to know when the SOAP message ends.

The following points can be noted on the SOAP envelope element.

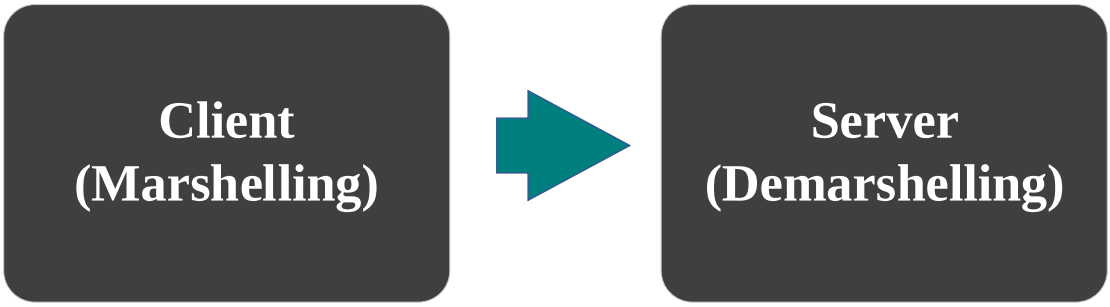
- Every SOAP message needs to have a root Envelope element. It is absolutely mandatory for SOAP message to have an envelope element.
- Every Envelope element needs to have at least one soap body element.
- If an Envelope element contains a header element, it must contain no more than one, and it must appear as the first child of the Envelope, before the body element.
- The envelope changes when SOAP versions change.

The fault messages

When a request is made to a SOAP web service, the response returned can be of either 2 forms which are a successful response or an error response. When a success is generated, the response from the server will always be a SOAP message. But if SOAP faults are generated, they are returned as "HTTP 500" errors.

SOAP Communication Model

All communication by SOAP is done via the HTTP protocol. Prior to SOAP, a lot of web services used the standard RPC (Remote Procedure Call) style for communication. This was the simplest type of communication, but it had a lot of limitations.



1. The client would format the information regarding the procedure call and any arguments into a SOAP message and sends it to the server as part of an HTTP request. This process of encapsulating the data into a SOAP message was known as **Marshalling**.
2. The server would then unwrap the message sent by the client, see what the client requested for and then send the appropriate response back to the client as a SOAP message. The practice of unwrapping a request sent by the client is known as **Demarshalling**.