Unit V- Introduction to AJAX and Webservices

SOAP

Simple Object Access Protocol (SOAP) is a protocol based on XML. It is used by the web services for exchange of information. It is a W3C recommendation.

This exchange of information is done over HTTP. This is a platform and language independent protocol.

The client-server communication is based on RPC. The HTTP does not designed to handle the distributed objects that are required by the RPC. Hence another application protocol is build over HTTP which popularly known as SOAP. SOAP allows to talk different applications that are running in two different operating systems.

Structure of SOAP

The structure of the SOAP is defined by four building blocks

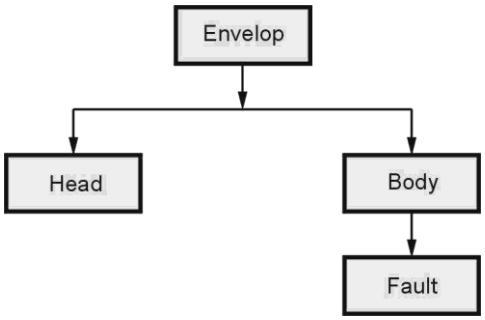

Fig. 10.6.1 SOAP building blocks

1.Envelop

The Envelop is the root of any SOAP document. It identifies that the XML document as a SOAP message. It defines two things - **Namespace** and **encodingStyle**. The Namespace value can be given as -

xmlns:soap="http://www.w3.org/2001/12/soap-envelope".

The encoding style is useful to define the data types used in the document. For

example the value of encoding can be given as -

soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding"

2.Header

The SOAP header field is optional. It contains the header information. The header contains three attributes such as **mustUnderstand, actor** and **encodingStyle**

The **mustUnderstand** attribute specifies 0 or 1. If the value is 1 then it means that the receiver should process the header.

Normally the SOAP message travels over a message path having the endpoints at the two ends.

The SOAP **actor** is used to specify the URI for this endpoint.

The **encodingStyle** is useful to define the data types used in the document

The SOAP header defines how the recipient should process the SOAP message.

3.Body

The required SOAP Body element contains the actual SOAP message intended for the ultimate endpoint of the message.

4.Fault

This is optional element of the SOAP message. It is used to represent the error code. It consists of elements such as **faultcode** which is a code for fault. The **faultstring** message gives the details about the fault.

SOAP and HTTP

HTTP communicates over TCP/IP. HTTP is based on Request-Response mechanism. SOAP is a HTTP Request-Response protocol. SOAP is a combination of HTTP and XML.

The SOAP request can be HTTP GET or HTTP POST request. Each request consists of two HTTP headers :**content-type** and **content-length**. The content-type specifies the MIME type and the content-length specifies the size of the message in bytes. After this header the body part begins which defines SOAP Envelop, Header, Body and Fault.

Example

The SOAP request can be

```
POST /mycreation HTTP/1.0 Host: www.mywebsite.com
Content-Type: text/xml; charset=utf-8 Content-Length: 500
<?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:Body xmlns:m="http://www.mywebsite.com/mycreation">
<m:GetDetails>
<m:Name>ABCD</m:Name>
</m:GetDetails>
</SOAP-ENV:Body>
</SOAP-
NV:Envelope>
```

The SOAP response can be

```
HTTP/1.0 200 OK
Content-Type:          text/xml;
charset=utf-8    Content-Length:
500
<?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:Body xmlns:m="http://www.mywebsite.com/mycreation">
      <m:GetDetailsResponse>
         <m:ItemDetails>This is a my new creation</m:ItemDetails>
      </m:GetDetailResponse>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>
```

SOAP Encoding

SOAP makes use of XML Schema to encode the programming constructs. Following are examples of SOAP encoding.

1. Encoding of Struct Data

Following is a code fragment which can be encoded as follows -

```
<xs:element name="Student">
<xs:complexType>
  <xs:sequence>
  <xs:element name="Name" type="xs:string"/>
  <xs:element name="RollNo" type="xs:integer"/>
   <xs:element name="address" type="tns:Address"/>
  </xs:sequence>
  <xs:element name="Address">
 <xs:complexType>
   < xs:sequence>
```

```
    < xs:element name="street" type="xsd:string"/>
    < xs:element name="city" type="xsd:string"/>
    < xs:element name="country" type="xsd:string"/>
</ xs:sequence>
</xs:complexTyp
e>
</e:element>
```

The following is an example of how the preceding schema definition could be subsequently used in a SOAP encoding.

```
<e:Student>
  <Name>AAA</Name>
  <RollNo>11</RollNo>
 <address>
            <street>XYZ Road</street>
            <city>PQR City</city>
            <country>India</country>
</address>
</e:Student
>
```

2. Encoding of Arrays

An array in SOAP is of type **SOAP-ENC:Array**

```
<element
 name="myArray"
 type="SOAP-
 ENC:Array"/>
```

The preceding schema can be represented in SOAP encoding as follows -

```
<myArray  SOAP-ENC:arrayType="xsd:int[3]">
  <number>1</number>
  <number>2</number>
  <number>3</number>
</myArray>
```