

DEVELOPING WEB SERVICES

SOAP

SOAP

Simple Object Access Protocol

- SOAP
 - ~~Simple Object Access Protocol~~
- SOAP is an open-standard, XML-based messaging **protocol** for exchanging information among computers. (Web Service!!)
 - As a protocol, there is a set of rules governing the exchange or transmission of data
- SOAP messages are written entirely in XML
- SOAP is a W3C recommendation

SYNTAX RULES

SOAP XML Syntax

- Must be encoded using XML
- Must use the SOAP Envelope namespace
 - `xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"`
- Must use the SOAP Encoding namespace
 - `soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding"`
- Must NOT contain a DTD reference
- Must NOT contain XML Processing Instructions

SOAP MESSAGE

Basic Format

```
<?xml version="1.0"?>
```

```
<soap:Envelope  
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"  
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
```

```
<soap:Header>
```

```
...
```

```
</soap:Header>
```

```
<soap:Body>
```

```
...
```

```
  <soap:Fault>
```

```
    ...
```

```
  </soap:Fault>
```

```
</soap:Body>
```

```
</soap:Envelope>
```


SOAP MESSAGE

Example - Request

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
  <soap:Body xmlns:m="http://www.website.com/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

SOAP MESSAGE

Example - Request

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
  <soap:Body xmlns:m="http://www.website.com/stock">
    <m:GetStockPriceResponse>
      <m:Price>192.42</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>
</soap:Envelope>
```

BUILDING BLOCKS

XML elements in SOAP message

- The root element **Envelope**
 - Used to Identify the XML document as a SOAP message
- Child element Header
 - Optional - Contains header information
- Child element Body
 - Contains all the requests and response information
 - Body child element Fault
 - Optional - Used only if a fault occurs

ENVELOPE

`<soap:Envelope>...</soap:Envelope>`

- Every SOAP message has a root Envelope element.
- Envelope is a mandatory part of SOAP message.
- Every Envelope element must contain exactly one Body element.
- If an Envelope contains a Header element,
 - it must appear as the first child of the Envelope, before the Body
 - Multiple headers may be included
- The SOAP envelope is specified using the namespace prefix and the Envelope element.
- The optional SOAP encoding is also specified using a namespace name and the optional encodingStyle element, which could also point to an encoding style other than the SOAP one.

HEADER

`<soap:Header>...</soap:Header>`

- Header element is optional
- Header elements can occur multiple times
- Headers are intended to add new features and functionality.
- The header is encoded as the first immediate child element of the SOAP envelope
- Header possible attributes
 - role
 - mustUnderstand
 - encodingStyle
 - relay

SOAP MESSAGE

Example - Header

...

<soap:Header

 mustUnderstand="true"

 relay="true"

 role="http://www.w3.org/2003/05/soap-envelope/role/
ultimateReceiver"

>

...

</soap:Header>

...

SOAP ROLES

Predefined SOAP Roles

- ultimateReceiver
 - It is the "web service", the one who actually processes the SOAP message.
 - <http://www.w3.org/2003/05/soap-envelope/role/ultimateReceiver>
- next
 - intermediary and ultimate receivers
 - <http://www.w3.org/2003/05/soap-envelope/role/next>
- none
 - Special role, no node should act as a none
 - either unprocessed or processing some other header block
 - <http://www.w3.org/2003/05/soap-envelope/role/none>

BODY

`<soap:Body>...</soap:Body>`

- Mandatory element that contains the main part to be proceed,
- Can contain any xml, as long as it's valid,
- Text should be nested inside a child element
- Recommend to name space qualify Body element children

SOAP MESSAGE

Example - Body

...

```
<soap:Body xmlns:m="http://www.website.com/stock">  
  <m:GetStockPriceResponse>  
    <m:Price>192.42</m:Price>  
  </m:GetStockPriceResponse>  
</soap:Body>
```

...

FAULT ELEMENT

<Code><Value> values

- Fault element is optional and used to indicate error messages
- If a Fault element is present, it must appear as a child element of the Body element.
- A Fault element can only appear once in a SOAP message.

SOAP MESSAGE

Example - Fault

```
...<soap:Body ... >
  <soap:Fault>
    <soap:Code>
      <soap:Value>soap:Sender</soap:Value>
      <soap:Subcode>
        <soap:Value>soap:Sender</soap:Value>
        <soap:Subcode><-- Recursive possible subcodes --></soap:Subcode>
      </soap:Subcode>
    </soap:Code>
    <soap:Reason>
      <soap:Text>Input Data Error</soap:Text>
    </soap:Reason>
    <soap:Node>
      <soap:Text>http://website.com/theNodeThatFailed</soap:Text>
    </soap:Node>
    <soap:Role>http://www.w3.org/2003/05/soap-envelope/role/ultimateReceiver</soap:Role>
  </soap:Fault>
</soap:Body>...
```

FAULT ELEMENT

<Code><Value> values

- VersionMismatch
 - The node that reported this error found a root element in the SOAP message that was not a valid Envelope element
- MustUnderstand
 - The node returning this fault did not understand a certain header child element
- DataEncodingUnknown
 - The node returning the fault did not understand the header element encoding (encodingStyle attribute) it was supposed to understand (the header was targeted at the node).
- Sender
 - The message was incorrectly formed or does not contain valid data, or lacks data etc.
 - (The sender sent a bad SOAP message)
- Receiver
 - The receiver of the SOAP message failed to process the message.