

SOAP Web Services

Laboratory of Service Design and Engineering

2011/2012



Outline

- What is SOAP?
- Web Services with SOAP
- WSDL
- Client Implementation
- Exercise



Simple Object Access Protocol



What is SOAP?

1. SOAP stands for Simple Object Access Protocol
2. SOAP is essentially an XML-based protocol for invoking remote methods.
3. SOAP is a protocol for accessing a Web Service.
4. SOAP is a format for sending messages
5. SOAP communicates via Internet
6. SOAP is platform independent
7. SOAP is language independent



Simple Object Access Protocol

- A SOAP message is an ordinary XML document containing the following elements:
 - An Envelope element that identifies the XML document as a SOAP message
 - A Header element that contains header information
 - A Body element that contains call and response information
 - A Fault element containing errors and status information

Skeleton SOAP Message

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

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

  <soap:Body>
    ...
    <soap:Fault>
      ...
    </soap:Fault>
  </soap:Body>

</soap:Envelope>
```


A SOAP Request Example

SOAP Request

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

  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>

</soap:Envelope>
```

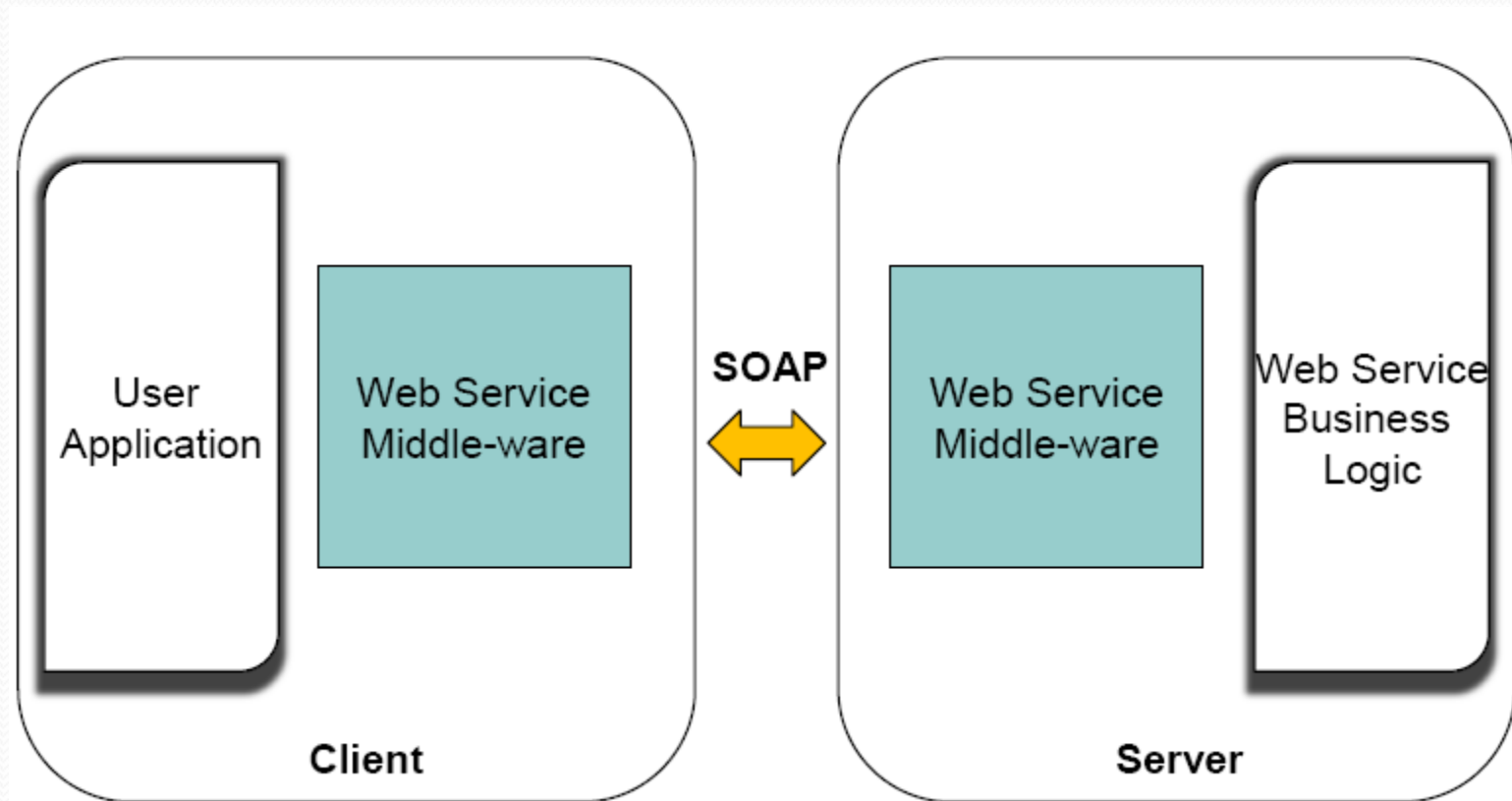
SOAP Response Example

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

  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>

</soap:Envelope>
```


Web Service Middle-Ware





Invoking JWS Service

- Two options
 - Using Dynamic Invocation Interface (DII)
 - Using Stubs generated from service WSDL description

Client Code: DII Method

```
import org.apache.axis.client.Service;
import org.apache.axis.client.Call;
import javax.xml.namespace.QName;

public class TestAddFunction {
    public static void main(String[] args) {
        try {
            String endpoint = "http://localhost:8080/axis/AddFunction.jws";
            Service service = new Service();
            Call call = (Call) service.createCall();
            call.setOperationName(new QName(endpoint, "addInt"));
            call.setTargetEndpointAddress(new java.net.URL(endpoint));
            Integer ret = (Integer) call.invoke(new Object[] { new
Integer(5), new Integer(6) });
            System.out.println("addInt(5, 6) = " + ret);
        } catch (Exception e) {
            System.err.println("Execution failed. Exception: " + e);
        }
    }
}
```




Web Service Description Language



WSDL

- WSDL stands for Web Services Description Language.
- WSDL is a document written in XML.
- The document describes a Web service.
- It specifies the location of the service and the operations (or methods) the service exposes.

WSDL

- A WSDL document describes a web service using these major elements:
- **<types>**
 - The data types used by the web service
- **<message>**
 - The messages used by the web service
- **<portType>**
 - The operations performed by the web service
- **<binding>**
 - The communication protocols used by the web service

WSDL Example

```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```


Example Service

- Length Convertor Service

 WebserviceX.NET

[Home](#) | [Web Services](#) | [UDDI Explorer](#) | [Tools](#) |

Top Web Services

[Stock Quote](#)
[Currency Convertor](#)
[Global Weather](#)
[London Gold And Silver Fixing](#)
[Real Time Market Data](#)
[SendSMSWorld](#)
[OFAC SDN and Blocked Persons](#)
[Translation Engine](#)
[ABA Lookup](#)
[SendSMSIndia](#)
[More..](#)

Web services enable to quickly integrate applications across multiple platforms, systems and even across businesses. Emerging standards such as SOAP, WSDL and UDDI will enable system-to-system communication that is easier and cheaper than traditional methods.

WebserviceX.NET provides programmable business logic components and standing data that serve as "black boxes" to provide functionality and data via web services.

WebserviceX.NET serves 6,000,000+ web services transactions every day

Browse web services by category - 70+ web services available now !

Business and Commerce	Communications
Standards and Lookup Data	Graphics and Multimedia
Other Web Services	Utilities
Value Manipulation / Unit Convertor	



Length Convertor

- What we need
 - A service (Already implemented at [websvc.net](http://www.websvc.net/))
 - Service WSDL
 - Download it <http://www.websvc.net/length.asmx?WSDL>
 - Stubs
 - Generate stubs from WSDL
 - Write client using stubs
 - Access the service (pass parameter and get results)



Lets do it



Exercise

- Access the Currency Convertor Service
 - Download WSDL file from
 - <http://www.websvcx.net/CurrencyConvertor.asmx?WSDL>
- Generate stubs using WSDL file.
 - Use given build.xml
- Create Client using stubs.
- Convert any currency to other currency type.