

Web Services

Web Services with AXIS

Javlon Eraliyev

Linz, Austria
2013

Contents

- 1 Web Services
 - Concept
 - SOAP
 - WSDL
 - UDDI
 - Overview

- 2 Web Services with AXIS
 - Installation and Usage
 - Examples

Web Service

is a method of communication between two electronic devices over the World Wide Web.

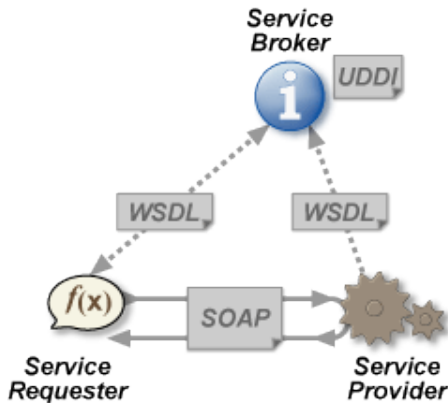


Figure : Web services architecture

SOAP = XML + HTTP + Standards

Simple Object Access Protocol

- SOAP is a method of transferring messages, or small amounts of information, over the Internet.
- SOAP messages are formatted in XML and are typically sent using HTTP
- For example, a user can send a SOAP message from a Windows machine to a Unix-based Web server without worrying about the message being altered

Example: SOAP

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

WSDL

Web Services Description Language

WSDL is an XML-based interface description language that is used for describing the functionality offered by a web service.

Describes:

- How the service can be called
- What parameters it expects
- What data structures it returns

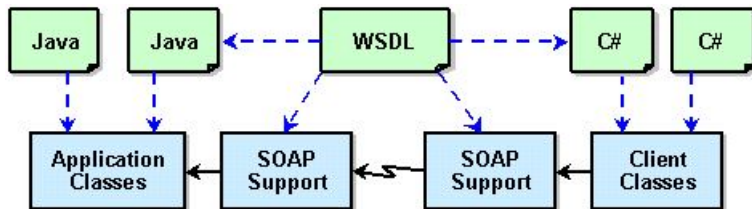


Figure : How two different languages can deal

Example: WSDL

```
<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>
```

UDDI

Universal Description Discovery and Integration = XML-based registry

- Provides a standardized method for publishing and discovering information about web services.
- UDDI registries are used in an enterprise to share Web Services

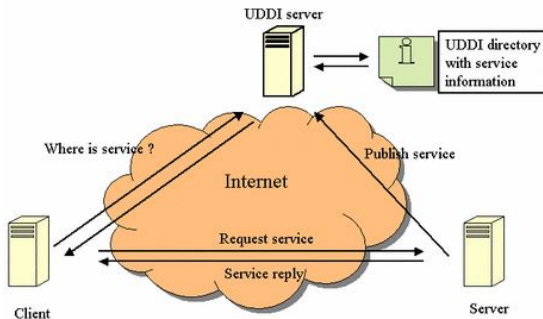
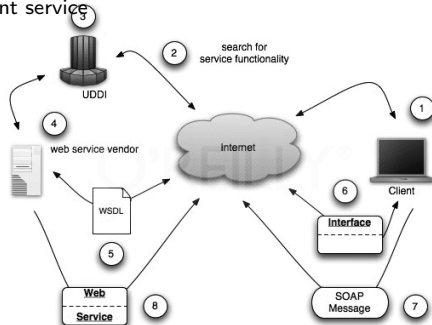


Figure : Service lookup using UDDI

Overview

- [1] Seek out service;
- [2] Connect to the directory to discover a relevant service;
- [3] Determine the presence of a service;
- [4] Check on availability and validity of vendor;
- [5] Vendor sends the client a WSDL;
- [6] Instance of the WS
- [7] SOAP request
- [8] Return values, responses



Example: WSDL

```
<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>
```

AXIS

Axis is essentially a SOAP engine – a framework for constructing SOAP processors such as clients, servers, gateways, etc. The current version of Axis is written in Java.

It also includes:

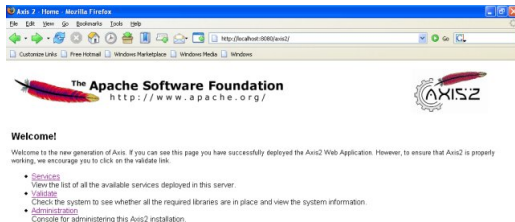
- a simple stand-alone server
- a server which plugs into servlet engines such as Tomcat
- extensive support for the WSDL
- emitter tooling that generates Java classes from WSDL
- a tool for monitoring TCP/IP packets

Installation and Usage

Apache Axis2 Installation Guide and Usage

- Download and install a Java Development Kit (JDK)
- Download and unpack the Axis2 Standard Binary Distribution and set an environment variable AXIS2_HOME
- Download and install Jakarta Tomcat
- Build Axis2 WAR file and drop it in the webapps directory of Tomcat
- Test it by pointing the web browser to the `http://<localhost:8080>/axis2`.

It should produce the following page



Examples

Create a Service Class

A StockQuoteService example seems to be mandatory in instances like this one, so let's use the following:

```
package samples.quickstart.service.pojo;  
  
import java.util.HashMap;  
  
public class StockQuoteService {  
    private HashMap map = new HashMap();  
  
    public double getPrice(String symbol) {  
        Double price = (Double) map.get(symbol);  
        if (price != null){  
            return price.doubleValue();  
        }  
        return 42.00;  
    }  
  
    public void update(String symbol, double price) {  
        map.put(symbol, new Double(price));  
    }  
}
```



Examples

Generate a WSDL file from a Java class

We need a WSDL file for our service. Axis2's Java2WSDL can be used to bootstrap a WSDL.

(Windows)

```
%AXIS2_HOME%\bin\java2wsdl -cp . -cn  
samples.quickstart.service.pojo.StockQuoteService -of StockQuoteService.wsdl
```

(Linux)

```
$AXIS2_HOME/bin/java2wsdl -cp . -cn  
samples.quickstart.service.pojo.StockQuoteService -of StockQuoteService.wsdl
```

Examples

Structure of StockQuoteService

The structure of this service will be as follows:

```
- StockQuoteService
  - META-INF
    - services.xml
  - lib
  - samples
    - quickstart
      - service
        - pojo
          - StockQuoteService.class
```

Examples

WSDL file of StockQuoteService

Genereted Service Definition File:

```
<service name="StockQuoteService" scope="application">
  <description>
    Stock Quote Sample Service
  </description>
  <messageReceivers>
    <messageReceiver
      mep="http://www.w3.org/2004/08/wsdl/in-only"
      class="org.apache.axis2.rpc.receivers.RPCInOnlyMessageReceiver"/>
    <messageReceiver
      mep="http://www.w3.org/2004/08/wsdl/in-out"
      class="org.apache.axis2.rpc.receivers.RPCMessageReceiver"/>
  </messageReceivers>
  <parameter name="ServiceClass">
    samples.quickstart.service.pojo.StockQuoteService
  </parameter>
</service>
```