Calling a SOAP Service (Axis2)

Since Object DataSource supports arbitrary JavaScript, it is quite simple to call any external APIs to obtain data. As an example, here is a SOAP client that invokes the **getStockQuote** service that is in the default configuration supplied by Apache Axis2. (http://axis.apache.org/axis2/java/core/docs/userguide.html)

Webservice Setup

1. Download **WAR** distribution from http://axis.apache.org/axis2/java/core/download.cgi
2. Deploy the WAR file to tomcat or any other web application
- tomcat download: http://tomcat.apache.org/
- copy **axis2.war** file to *tomcat/webapps*
3. Start Tomcat server
4. Login at *http://localhost:8087/axis2/axis2-admin/*
- port number: refer to */tomcat/conf/server.xml*
- login details, refer to http://axis.apache.org/axis2/java/core/docs/webadminguide.html#login
5. Upload webservice, **StockQuoteServer.jar** (attached)

Repertoire Setup

6. Download BIN distribution from http://axis.apache.org/axis2/java/core/download.cgi
7. Create a folder name **Axis2** in *\\Repertoire\ext\*
8. Copy ALL .JAR files from *axis2-x.X.X\lib* to *\\Repertoire\ext\Axis2*
9. Start Elixir Repertoire

For more information on the Setup, please refer to:
http://axis.apache.org/axis2/java/core/docs/installationguide.html

Creating an Object Datasource

10. Create a new Object Datasource in Repertoire, **GetStockQuote.ds** (see attached for the Ready copy). You are able to write Javascript in Object datasource to retrieve data from the web service.
- Add one field to the schema definition - Price - with type Float. Click **Next**.
- Refer to **AXIOMClient.java** for sample code which is a direct conversion for **GetStockQuote.ds**

OR
- Copy and Paste the following code into the JavaScript editor:

```
importClass(org.apache.axiom.om.OMAbstractFactory);
importClass(org.apache.axiom.om.OMElement);
importClass(org.apache.axiom.om.OMFactory);
importClass(org.apache.axiom.om.OMNamespace);
importClass(org.apache.axis2.Constants);
importClass(org.apache.axis2.addressing.EndpointReference);
importClass(org.apache.axis2.client.Options);
importClass(org.apache.axis2.client.ServiceClient);

function pushTo(/*PushContext*/ cxt, /*DataListener*/ dl)
{
  // call soap
var fac = new org.apache.axiom.om.OMAbstractFactory.getOMFactory();
var omNs = fac.createOMNamespace("http://quickstart.samples/xsd", "tns");
var getPricePayload = fac.createOMElement("getPrice", omNs);
var value = fac.createOMElement("symbol", omNs);

value.addChild(fac.createOMText(value, "IBM"));
getPricePayload.addChild(value);

var pRef = "http://localhost:8087/axis2/services/StockQuoteService";

var options = new Packages.org.apache.axis2.client.Options();
var endpointReference = new Packages.org.apache.axis2.addressing.EndpointReference(pRef);

options.setTo(endpointReference);
options.setTransportInProtocol(Constants.TRANSPORT_HTTP);

var sender = new Packages.org.apache.axis2.client.ServiceClient();
sender.setOptions(options);

var result = sender.sendReceive(getPricePayload);
sender.cleanupTransport();

  // now send the record
  dl.startData(this);
  rec = this.newRecordInstance();
  data = rec.getData();
  data[0] = result.getFirstElement().getText();
  dl.processRecord(rec);
  dl.endData(this);
}
```

- Finish the wizard and then **Save**.
- Finally, click **Load Data**. You should see a single record with a single field (Price) produced.
The Data should display as: 42.0

You can now experiment with SOAP services that supply more records and/or fields.
If you pass parameters into the invocation (eg. **call.invoke(new Array("1","2")**) , you can use

dynamic substitution to pass the values in from outside:

```
call.invoke(new Array("${First}","${Second}"));
```