WS - Proof of Concept 2007-10-2007

Unable to render {include} The included page could not be found.

Summary

12/10/2007 A proof of concept of Axis2 web services was completed. It is a service that takes a query for a subject search and returns the subjects. It uses the same API that the UI uses to search for subjects. The Axis2 web service supports SOAP and REST. The proof of concept is not an actual service that we will use, the inputs and outputs and design of which services we need will be more thoroughly thought out, the intent is to get Axis up and running. Also, I cleansed all the data, so if you notice that it is not consistent, it is due to data cleansing.

The Service

The service is a method which takes a query for subjects, and returns an array of subjects (at some point I had lists working with Axis, then it stopped working, so I am left with an array)

```
package edu.internet2.middleware.grouper.webservices;

import java.util.Set;

import edu.internet2.middleware.grouper.SubjectFinder;
import edu.internet2.middleware.subject.provider.JDBCSubject;

public class GrouperService {

	/**
	 * web service wrapper for find all subjects based on query
	 * @param query
	 * @return
	 */
	@SuppressWarnings("unchecked")
	public WsSubject[] findAll(String query) {
		Set<JDBCSubject> subjectSet = SubjectFinder.findAll(query);
		if (subjectSet == null || subjectSet.size() == 0) {
			return null;
		}
		//convert the set to a list
		WsSubject[] results = new WsSubject[subjectSet.size()];
		int i=0;
		for (JDBCSubject jdbcSubject : subjectSet) {
			WsSubject wsSubject = new WsSubject(jdbcSubject);
			results[i++] = wsSubject;
		}
		return results;
	}

}
```

Note that the return type is WsSubject (detailed below). I am assuming that the beans used for the web service will be new, and used solely for the web services. The WSDL is based (automatically) on the GrouperService class and the beans it uses. Here is the bean:

```
package edu.internet2.middleware.grouper.webservices;

import edu.internet2.middleware.subject.provider.JDBCSubject;

/**
 * subject bean for web services
 * @author mchyzer
 *
 */
public class WsSubject {

	public WsSubject(){}

	public WsSubject(JDBCSubject jdbcSubject) {
		this.id = jdbcSubject.getId();
		this.description = jdbcSubject.getDescription();
		this.name = jdbcSubject.getName();
	}

	private String id;
	private String name;
	private String description;
	/**
	 * @return the id
	 */
	public String getId() {
		return id;
	}
	/**
	 * @param id the id to set
	 */
	public void setId(String id) {
		this.id = id;
	}
	/**
	 * @return the name
	 */
	public String getName() {
		return name;
	}
	/**
	 * @param name the name to set
	 */
	public void setName(String name) {
		this.name = name;
	}
	/**
	 * @return the description
	 */
	public String getDescription() {
		return description;
	}
	/**
	 * @param description the description to set
	 */
	public void setDescription(String description) {
		this.description = description;
	}

}
```

REST Results

When I go to this url: http://localhost:8090/grouper/services/GrouperService/findAll?query=100abcd, I see this (well, after adding some whitespace):

```
<ns:findAllResponse>
  <ns:return type="edu.internet2.middleware.grouper.webservices.WsSubject">
    <ns:description>JUSTIN SMITH</ns:description>
    <ns:id>100abcda</ns:id>
    <ns:name>JUSTIN SMITH</ns:name>
  </ns:return>
  <ns:return type="edu.internet2.middleware.grouper.webservices.WsSubject">
    <ns:description>MICHAEL SMITH</ns:description>
    <ns:id>100abcdb</ns:id>
    <ns:name>MICHAEL SMITH</ns:name>
    </ns:return>
  <ns:return type="edu.internet2.middleware.grouper.webservices.WsSubject">
    <ns:description>MARLENE SMITH</ns:description>
    <ns:id>100abcdc</ns:id>
    <ns:name>MARLENE SMITH</ns:name>
  </ns:return>
</ns:findAllResponse>
```

Here is a Java REST client which does not use any Java data structures that the SOAP client uses:

```
package edu.internet2.middleware.grouper.webservicesClient;

import java.io.Reader;
import java.io.StringReader;
import java.util.List;

import org.apache.commons.httpclient.HttpClient;
import org.apache.commons.httpclient.methods.GetMethod;
import org.jdom.Document;
import org.jdom.Element;
import org.jdom.input.SAXBuilder;

public class RunGrouperServiceNonAxis {

        /**
         * @param args
         */
        @SuppressWarnings("unchecked")
        public static void main(String[] args) throws Exception {
                HttpClient httpClient = new HttpClient();
                GetMethod getMethod = new GetMethod(
                                "http://localhost:8090/grouper/services/GrouperService/findAll?query=1002136");
                httpClient.executeMethod(getMethod);
                int statusCode = getMethod.getStatusCode();
                // see if request worked or not
                if (statusCode != 200) {
                        throw new RuntimeException("Bad response from web service: "
                                        + statusCode);
                }
                String response = getMethod.getResponseBodyAsString();
                Reader xmlReader = new StringReader(response);
                try {
                        // process xml
                        Document document = new SAXBuilder().build(xmlReader);
                        Element findAllResponse = document.getRootElement();
                        assertTrue("findAllResponse".equals(findAllResponse.getName()),
                                        "root not findAllResponse: " + findAllResponse.getName());

                        // process each child
                        List<Element> returns = findAllResponse.getChildren();
                        for (Element theReturn : returns) {

                                // make sure they have the right element name and type
                                assertTrue("return".equals(theReturn.getName()),
                                                "element not 'return': " + theReturn.getName());
                                assertTrue(
                                                "edu.internet2.middleware.grouper.webservices.WsSubject"
                                                                .equals(theReturn.getAttributeValue("type")),
```

```
                                        "type is: " + theReturn.getAttributeValue("type"));

                                String description = theReturn.getChild("description",
                                        findAllResponse.getNamespace()).getValue();
                                String id = theReturn.getChild("id",
                                        findAllResponse.getNamespace()).getValue();
                                String name = theReturn.getChild("name",
                                        findAllResponse.getNamespace()).getValue();

                                System.out.println(id + " - " + name + " - " + description);

                        }
                } finally {
                        try {
                                xmlReader.close();
                        } catch (Exception e) {
                        }
                }

        }

        /**
         * assert like java 1.4 assert
         *
         * @param isTrue
         * @param reason
         */
        private static void assertTrue(boolean isTrue, String reason) {
                if (!isTrue) {
                        throw new RuntimeException(reason);
                }
        }

}
```

SOAP Results

First need to make a SOAP client. Before we can do that we need the WSDL from here (Axis automatically generated this WSDL): http://localhost:8090/grouper/services/GrouperService?wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:axis2="http://webservices.grouper.
middleware.internet2.edu/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:ns0="http://webservices.
grouper.middleware.internet2.edu/xsd" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:http="
http://schemas.xmlsoap.org/wsdl/http/" xmlns:ns1="http://org.apache.axis2/xsd" xmlns:wsaw="http://www.w3.org
/2006/05/addressing/wsdl" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org
/wsdl/soap/" targetNamespace="http://webservices.grouper.middleware.internet2.edu/">
    <wsdl:documentation>GrouperService</wsdl:documentation>
    <wsdl:types>
        <xs:schema xmlns:ns="http://webservices.grouper.middleware.internet2.edu/xsd" attributeFormDefault="
qualified" elementFormDefault="qualified" targetNamespace="http://webservices.grouper.middleware.internet2.edu
/xsd">
            <xs:element name="findAll">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="query" nillable="true" type="xs:string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="findAllResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element maxOccurs="unbounded" minOccurs="0" name="return" nillable="true" type="ns:
WsSubject"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:complexType name="WsSubject">
                <xs:sequence>
```

```
                    <xs:element minOccurs="0" name="description" nillable="true" type="xs:string"/>
                    <xs:element minOccurs="0" name="id" nillable="true" type="xs:string"/>
                    <xs:element minOccurs="0" name="name" nillable="true" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
        </xs:schema>
    </wsdl:types>
    <wsdl:message name="findAllRequest">
        <wsdl:part name="parameters" element="ns0:findAll"/>
    </wsdl:message>
    <wsdl:message name="findAllResponse">
        <wsdl:part name="parameters" element="ns0:findAllResponse"/>
    </wsdl:message>
    <wsdl:portType name="GrouperServicePortType">
        <wsdl:operation name="findAll">
            <wsdl:input message="axis2:findAllRequest" wsaw:Action="urn:findAll"/>
            <wsdl:output message="axis2:findAllResponse" wsaw:Action="urn:findAllResponse"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="GrouperServiceSOAP11Binding" type="axis2:GrouperServicePortType">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
        <wsdl:operation name="findAll">
            <soap:operation soapAction="urn:findAll" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="GrouperServiceSOAP12Binding" type="axis2:GrouperServicePortType">
        <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
        <wsdl:operation name="findAll">
            <soap12:operation soapAction="urn:findAll" style="document"/>
            <wsdl:input>
                <soap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="GrouperServiceHttpBinding" type="axis2:GrouperServicePortType">
        <http:binding verb="POST"/>
        <wsdl:operation name="findAll">
            <http:operation location="GrouperService/findAll"/>
            <wsdl:input>
                <mime:content type="text/xml" part="findAll"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="text/xml" part="findAll"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="GrouperService">
        <wsdl:port name="GrouperServiceSOAP11port_http" binding="axis2:GrouperServiceSOAP11Binding">
            <soap:address location="http://localhost:8090/grouper/services/GrouperService"/>
        </wsdl:port>
        <wsdl:port name="GrouperServiceSOAP12port_http" binding="axis2:GrouperServiceSOAP12Binding">
            <soap12:address location="http://localhost:8090/grouper/services/GrouperService"/>
        </wsdl:port>
        <wsdl:port name="GrouperServiceHttpport" binding="axis2:GrouperServiceHttpBinding">
            <http:address location="http://localhost:8090/grouper/services/GrouperService"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

Then we can use Axis to make a Java SOAP client:

```
C:\mchyzer\isc\dev\grouper\axisJar>wsdl2java -p edu.internet2.middleware.grouper.webservicesClient -t -uri
GrouperService.wsdl
```

Then we just make a call with very simple Java code (the underlying classes were generated by Axis with the above command):

```
/**
 *
 */
package edu.internet2.middleware.grouper.webservicesClient;

import edu.internet2.middleware.grouper.webservicesClient.GrouperServiceStub.FindAll;
import edu.internet2.middleware.grouper.webservicesClient.GrouperServiceStub.FindAllResponse;
import edu.internet2.middleware.grouper.webservicesClient.GrouperServiceStub.WsSubject;

/**
 * @author mchyzer
 *
 */
public class RunGrouperService {

	/**
	 * @param args
	 */
	public static void main(String[] args) throws Exception {
		GrouperServiceStub stub = new GrouperServiceStub(
				"http://localhost:8091/grouper/services/GrouperService");

		FindAll findAll = new FindAll();
		findAll.setQuery("1002136");
		FindAllResponse findAllResponse = stub.findAll(findAll);
		WsSubject[] subjects = findAllResponse.get_return();
		if (subjects == null) {
			System.out.println("No results");
		} else {
			for (WsSubject wsSubject : subjects) {
				System.out.println(wsSubject.getId() + " - "
					+ wsSubject.getName() + " - "
					+ wsSubject.getDescription());
			}
		}

	}

}
```

Here are the results:

```
100abcda - MICHAEL SMITH - MICHAEL SMITH
100abcdb - JUSTIN SMITH - JUSTIN SMITH
100abcdc - MARLENE D SMITH - MARLENE D SMITH
```

Here is the SOAP message that was transmitted:

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
<soapenv:Body>
  <ns1:findAll xmlns:ns1="http://webservices.grouper.middleware.internet2.edu/xsd">
    <ns1:query>100abcd</ns1:query>
  </ns1:findAll>
</soapenv:Body>
</soapenv:Envelope>
```

Here is the SOAP response:

```
<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
<soapenv:Body>
  <ns:findAllResponse xmlns:ns="http://webservices.grouper.middleware.internet2.edu/xsd">
    <ns:return type="edu.internet2.middleware.grouper.webservices.WsSubject">
      <ns:description>MICHAEL SMITH</ns:description>
      <ns:id>100abcda</ns:id>
      <ns:name>MICHAEL SMITH</ns:name>
    </ns:return>
    <ns:return type="edu.internet2.middleware.grouper.webservices.WsSubject">
      <ns:description>JUSTIN SMITH</ns:description>
      <ns:id>100abcdb</ns:id>
      <ns:name>JUSTIN SMITH</ns:name>
    </ns:return>
    <ns:return type="edu.internet2.middleware.grouper.webservices.WsSubject">
      <ns:description>MARLENE D SMITH</ns:description>
      <ns:id>100abcdc</ns:id>
      <ns:name>MARLENE D SMITH</ns:name>
    </ns:return>
  </ns:findAllResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Details

1. I downloaded the latest stable Axis2: v1.3
2. I made a directory with the contents, and set an env variable: AXIS2_HOME
3. I copied all the jars (~40) to grouper/lib, and removed the dupes (I think the only one was xml-apis.jar)
4. Copy the axis quick start directories: "services", "modules", "conf" to grouper-ui/webapp/WEB-INF
5. Create an axis archive (.aar) file. Based this on the Axis quickstart one:

```
C:\temp\grouperaar>find
.
.\edu
.\edu\internet2
.\edu\internet2\middleware
.\edu\internet2\middleware\grouper
.\edu\internet2\middleware\grouper\webservices
.\edu\internet2\middleware\grouper\webservices\GrouperService.class
.\edu\internet2\middleware\grouper\webservices\WsSubject.class
.\GrouperService.aar
.\META-INF
.\META-INF\MANIFEST.MF
.\META-INF\services.xml
```

Each time the service classes are changed (these are also in the grouper/src), the classfiles need to be put in the GrouperService.aar file. The services.xml file has the following contents:

```
<service name="GrouperService" scope="application" targetNamespace="http://webservices.grouper.middleware.
internet2.edu/">
    <description>
        Grouper 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>
    <schema schemaNamespace="http://webservices.grouper.middleware.internet2.edu/xsd"/>
    <parameter name="ServiceClass">edu.internet2.middleware.grouper.webservices.GrouperService</parameter>
</service>
```

Zip that up in GrouperSystem.aar, and put that in the WEB-INF/services dir

1. Do a "dist" ui build, start the app server

Questions or comments? Contact us.

Unable to render {include} The included page could not be found.