

Simple Web Service with JAX-WS

{scrollbar}

Application Overview application

The sample application referred in this article is a simple calculator which performs addition of two integers. The client application referred here is not a J2EE application. It is a regular Java client which will call a web service to carry out the application functionality. Web service is exposed as a Servlet in the Geronimo application server.

Service implementation

The **Calculator** interface defines the Service Endpoint Interface (SEI) for the Web Service.

```
javasolidCalculator.java package org.apache.geronimo.samples.jws; import javax.xml.ws.WebService; import javax.xml.ws.WebMethod; import javax.xml.ws.WebParam; @WebService(name="CalculatorPortType", targetNamespace = "http://jws.samples.geronimo.apache.org") public interface Calculator { @WebMethod public int add(@WebParam(name = "value1") int value1, @WebParam(name = "value2") int value2); }
```

The **CalculatorService** class implements the Web Service business logic. It implements all the methods defined in the SEI. The class does not need to implement the **Calculator** interface but must reference it through the `@WebService.endpointInterface` annotation. This class will be exposed as a Servlet through **web.xml** file even though it does not extend the **javax.servlet.Servlet** class.

The **context** variable marked with the `@Resource` annotation will be injected at runtime. The **WebServiceContext** can be used to obtain the message context and security information relative to the call.

```
javasolidCalculatorService.java package org.apache.geronimo.samples.jws; import javax.xml.annotation.Resource; import javax.xml.ws.WebService; import javax.xml.ws.WebServiceContext; @WebService(serviceName = "Calculator", portName="CalculatorPort", endpointInterface = "org.apache.geronimo.samples.jws.Calculator", targetNamespace = "http://jws.samples.geronimo.apache.org", wsdlLocation = "WEB-INF/wsdl/CalculatorService.wsdl") public class CalculatorService implements Calculator { @Resource private WebServiceContext context; public int add(int value1, int value2) { System.out.println("User Principal: " + context.getUserPrincipal()); return value1 + value2; } }
```

The **web.xml** descriptor is used to deploy the Web Service.

```
xmlsolidweb.xml <?xml version="1.0" encoding="UTF-8"?> <web-app xmlns="http://java.sun.com/xml/ns/javaee" version="2.5"> <servlet> <display-name>CalculatorService</display-name> <servlet-name>CalculatorService</servlet-name> <servlet-class> org.apache.geronimo.samples.jws.CalculatorService </servlet-class> </servlet> <servlet-mapping> <servlet-name>CalculatorService</servlet-name> <url-pattern>/calculator</url-pattern> </servlet-mapping> ..... </web-app> web.xml
```

The **web.xml** descriptor is not necessary for simple JAX-WS web service deployments. If the **web.xml** descriptor is not provided, it will be automatically generated during deployment.

The **geronimo-web.xml** descriptor is optional but it is used in this sample to specify the module name. Information about the project (e.g. module's unique identification, any dependencies) is described inside the `<environment>` tag. In this case, there are no dependencies so they do not need to be listed. However, it is a good idea to give this module some sort of unique identification, so that it can later be referenced by some other deployable application.

```
xmlsolidgeronimo-web.xml <?xml version="1.0" encoding="UTF-8"?> <web-app xmlns="http://geronimo.apache.org/xml/ns/j2ee/web-1.1"> <dep:environment xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.1"> <dep:moduleId> <dep:groupId>${pom.groupId}</dep:groupId> <dep:artifactId>${pom.artifactId}</dep:artifactId> <dep:version>${version}</dep:version> <dep:type>war</dep:type> </dep:environment> <context-root>/jaxws-calculator</context-root> <service-ref> <service-ref-name>services/Calculator</service-ref-name> <port> <port-name>CalculatorPort</port-name> <protocol>http</protocol> <host>localhost</host> <port>8080</port> <uri>/jaxws-calculator/calculator</uri> </port> </service-ref> </web-app>
```

The following WSDL file describes the Web Service:

```
xmlsolidCalculatorService.wsdl <?xml version="1.0" encoding="UTF-8"?> <wsdl:definitions name="Calculator" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://jws.samples.geronimo.apache.org" xmlns:tns="http://jws.samples.geronimo.apache.org"> <wsdl:types> <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://jws.samples.geronimo.apache.org" targetNamespace="http://jws.samples.geronimo.apache.org" attributeFormDefault="unqualified" elementFormDefault="qualified"> <xsd:element name="add"> <xsd:complexType> <xsd:sequence> <xsd:element name="value1" type="xsd:int"/> <xsd:element name="value2" type="xsd:int"/> </xsd:sequence> </xsd:complexType> </xsd:element> <xsd:element name="addResponse"> <xsd:complexType> <xsd:sequence> <xsd:element name="return" type="xsd:int"/> </xsd:sequence> </xsd:complexType> </xsd:element> </xsd:schema> </wsdl:types> <wsdl:message name="add"> <wsdl:part name="add" element="tns:add"/> </wsdl:message> <wsdl:message name="addResponse"> <wsdl:part name="addResponse" element="tns:addResponse"/> </wsdl:message> <wsdl:portType name="CalculatorPortType"> <wsdl:operation name="add"> <wsdl:input name="add" message="tns:add"/> <wsdl:output name="addResponse" message="tns:addResponse"/> </wsdl:operation> </wsdl:portType> <wsdl:binding name="CalculatorSoapBinding" type="tns:CalculatorPortType"> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="add"> <soap:operation soapAction="add" style="document"/> <wsdl:input name="add"> <soap:body use="literal"/> </wsdl:input> <wsdl:output name="addResponse"> <soap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding> <wsdl:service name="Calculator"> <wsdl:port name="CalculatorPort" binding="tns:CalculatorSoapBinding"> <soap:address location="http://localhost:8080/jaxws-calculator-1.0/calculator"/> </wsdl:port> </wsdl:service> </wsdl:definitions> webservices.xml
```

In J2EE version 1.4, the **webservices.xml** file was also necessary to describe the Web Service. In Java EE 5 that file is optional and is not required in this example.

JSP-based JAX-WS client

The **add.jsp** is a basic client for the **CalculatorService** Web Service.

```

javasolidadd.jsp <%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%> <%@ page import="javax.
naming.InitialContext,javax.xml.ws.Service,org.apache.geronimo.samples.jws.Calculator"%> <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head> <title>Apache Geronimo Sample Application - JAX-WS Calculator</title> <meta content="text/html; CHARSET=iso-8859-1" http-equiv="Content-
Type"> </head> <BODY> <font face="Verdana, Helvetica, Arial"> <h3>This is a JAX-WS web service sample application. Please type the value 1 and
value 2 below to see the add result.</h3> <form action="add.jsp"> Value 1: <input type="text" name="value1"> Value 2: <input type="text" name="value2"
> <input type="submit" value="Add"> </form> <br> <% String value1 = request.getParameter( "value1" ); String value2 = request.getParameter( "value2" );
if (value1 != null && value1.trim().length() > 0 && value2 != null && value2.trim().length() > 0) { out.println("<h4>"); try { int v1 = Integer.parseInt(value1); int
v2 = Integer.parseInt(value2); InitialContext ctx = new InitialContext(); Service service = (Service)ctx.lookup("java:comp/env/services/Calculator");
Calculator calc = service.getPort(Calculator.class); int sum = calc.add(v1, v2); out.println("Result: " + v1 + " + " + v2 + " = " + sum); } catch ( Exception e ) {
out.println("Error: " + e.getMessage()); } out.println("</h4>"); } %> </FONT> </body> </html>

```

The **add.jsp** looks up a Web Service reference in the JNDI tree. The Web Service reference must first be added the **web.xml** file.

```

xmldsolidweb.xml <?xml version="1.0" encoding="UTF-8"?> <web-app xmlns="http://java.sun.com/xml/ns/javaee" version="2.5"> ..... <service-ref> <service-
ref-name>services/Calculator</service-ref-name> <service-interface>javax.xml.ws.Service</service-interface> <wsdl-file>WEB-INF/wsdl
/CalculatorService.wsdl</wsdl-file> </service-ref> ..... </web-app>

```

Resource injection in JSP

Since the resource injection is not supported in JSPs the service-ref must be added explicitly to the **web.xml** file.

Building and Deploying Sample Application configure

Checkout the sample source code from SVN:

```

svn checkout http://svn.apache.org/repos/asf/geronimo/samples/trunk/samples/jaxws-calculator calculator

```

The source code will be checked out into the **calculator/** directory. This sample code code lives in **calculator/jaxws-calculator-war/** directory.

Required Tools

The tools used for developing and building the Calculator sample application are:

Apache Maven 2.0.x

[Apache Maven](#) is used for building the Calculator application.

Building

Compile Source Code

From command prompt execute the following command in the **calculator/jaxws-calculator-war/** folder:

```

mvn install

```

After the code is successfully compiled a **jaxws-calculator-war-2.0-SNAPSHOT.war** file will be created in the **target/** subfolder.

Deploying

Deploy the **jaxws-calculator-war-2.0-SNAPSHOT.war** using the Geronimo Console (<http://localhost:8080/console>):

1. Scroll to **Deploy New** from the **Console Navigation** panel.
2. Load **jaxws-calculator-war-2.0-SNAPSHOT.war** from **calculator/jaxws-calculator-war/target/** folder in to the **Archive** input box.
3. Press **Install** button to deploy application in the server.

Testing of the Sample testing

To test this sample service use the **add.jsp** (<http://localhost:8080/jaxws-calculator/add.jsp>) to invoke the Web Service. Once the JSP page loads type in two values to add and press the **Add** button. The result of the addition should show up below. For example:

