@WebService handlers with @HandlerChain

Example webservice-handlerchain can be browsed at https://github.com/apache/tomee/tree/master/examples/webservice-handlerchain

In this example we see a basic JAX-WS `@WebService` component use a handler chain to alter incoming and outgoing SOAP messages. SOAP Handlers are similar to Servlet Filters or EJB/CDI Interceptors.

At high level, the steps involved are:

1. Create handler(s) implementing `javax.xml.ws.handler.soap.SOAPHandler`
2. Declare and order them in an xml file via `<handler-chain>`
3. Associate the xml file with an `@WebService` component via `@HandlerChain`

The @HandlerChain

First we'll start with our plain `@WebService` bean, called `Calculator`, which is annotated with `@HandlerChain`

```
@Singleton
@WebService(
        portName = "CalculatorPort",
        serviceName = "CalculatorService",
        targetNamespace = "http://superbiz.org/wsdl",
        endpointInterface = "org.superbiz.calculator.wsh.CalculatorWs")
@HandlerChain(file = "handlers.xml")
public class Calculator implements CalculatorWs {

    public int sum(int add1, int add2) {
        return add1 + add2;
    }

    public int multiply(int mul1, int mul2) {
        return mul1 * mul2;
    }
}
```

Here we see `@HandlerChain` pointing to a file called `handlers.xml`. This file could be called anything, but it must be in the same jar and java package as our `Calculator` component.

The <handler-chains> file

Our `Calculator` service is in the package `org.superbiz.calculator.wsh`, which means our handler chain xml file must be at `org/superbiz/calculator/wsh/handlers.xml` in our application's classpath or

the file will not be found and no handlers will be used.

In maven we achieve this by putting our handlers.xml in `src/main/resources` like so:

- `src/main/resources/org/superbiz/calculator/wsh/handlers.xml`

With this file we declare and **order** our handler chain.

```
<?xml version="1.0" encoding="UTF-8"?>
<handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
  <handler-chain>
    <handler>
      <handler-name>org.superbiz.calculator.wsh.Inflate</handler-name>
      <handler-class>org.superbiz.calculator.wsh.Inflate</handler-class>
    </handler>
    <handler>
      <handler-name>org.superbiz.calculator.wsh.Increment</handler-name>
      <handler-class>org.superbiz.calculator.wsh.Increment</handler-class>
    </handler>
  </handler-chain>
</handler-chains>
```

The order as you might suspect is:

- `Inflate`
- `Increment`

The SOAPHandler implementation

Our `Inflate` handler has the job of monitoring **responses** to the `sum` and `multiply` operations and making them 1000 times better. Manipulation of the message is done by walking the `SOAPBody` and editing the nodes. The `handleMessage` method is invoked for both requests and responses, so it is important to check the `SOAPBody` before attempting to naviage the nodes.

```
import org.w3c.dom.Node;
import javax.xml.namespace.QName;
import javax.xml.soap.SOAPBody;
import javax.xml.soap.SOAPException;
import javax.xml.soap.SOAPMessage;
import javax.xml.ws.handler.MessageContext;
import javax.xml.ws.handler.soap.SOAPHandler;
import javax.xml.ws.handler.soap.SOAPMessageContext;
import java.util.Collections;
import java.util.Set;

public class Inflate implements SOAPHandler<SOAPMessageContext> {

    public boolean handleMessage(SOAPMessageContext mc) {
        try {
            final SOAPMessage message = mc.getMessage();
            final SOAPBody body = message.getSOAPBody();
            final String localName = body.getFirstChild().getLocalName();

            if ("sumResponse".equals(localName) || "multiplyResponse".equals(
localName)) {
                final Node responseNode = body.getFirstChild();
                final Node returnNode = responseNode.getFirstChild();
                final Node intNode = returnNode.getFirstChild();

                final int value = new Integer(intNode.getNodeValue());
                intNode.setNodeValue(Integer.toString(value * 1000));
            }

            return true;
        } catch (SOAPException e) {
            return false;
        }
    }

    public Set<QName> getHeaders() {
        return Collections.emptySet();
    }

    public void close(MessageContext mc) {
    }

    public boolean handleFault(SOAPMessageContext mc) {
        return true;
    }
}
```

The `Increment` handler is identical in code and therefore not shown. Instead of multiplying by 1000, it simply adds 1.

The TestCase

We use the JAX-WS API to create a Java client for our `Calculator` web service and use it to invoke both the `sum` and `multiply` operations. Note the clever use of math to assert both the existence and order of our handlers. If `Inflate` and `Increment` were reversed, the responses would be 11000 and 13000 respectively.

```
public class CalculatorTest {

    @BeforeClass
    public static void setUp() throws Exception {
        Properties properties = new Properties();
        properties.setProperty("openejb.embedded.remotable", "true");
        EJBContainer.createEJBContainer(properties);
    }

    @Test
    public void testCalculatorViaWsInterface() throws Exception {
        final Service calculatorService = Service.create(
                new URL("http://127.0.0.1:4204/Calculator?wsdl"),
                new QName("http://superbiz.org/wsdl", "CalculatorService"));

        assertNotNull(calculatorService);

        final CalculatorWs calculator = calculatorService.getPort(CalculatorWs.class);

        // we expect our answers to come back 1000 times better, plus one!
        assertEquals(10001, calculator.sum(4, 6));
        assertEquals(12001, calculator.multiply(3, 4));
    }
}
```

Running the example

Simply run `mvn clean install` and you should see output similar to the following:

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running org.superbiz.calculator.wsh.CalculatorTest
INFO - openejb.home = /Users/dblevins/work/all/trunk/openejb/examples/webservice-
handlers
INFO - openejb.base = /Users/dblevins/work/all/trunk/openejb/examples/webservice-
handlers
INFO - Using 'javax.ejb.embeddable.EJBContainer=true'
INFO - Cannot find the configuration file [conf/openejb.xml].  Will attempt to create
one for the beans deployed.
```

```
INFO - Configuring Service(id=Default Security Service, type=SecurityService,
provider-id=Default Security Service)
INFO - Configuring Service(id=Default Transaction Manager, type=TransactionManager,
provider-id=Default Transaction Manager)
INFO - Creating TransactionManager(id=Default Transaction Manager)
INFO - Creating SecurityService(id=Default Security Service)
INFO - Beginning load: /Users/dblevins/work/all/trunk/openejb/examples/webservice-
handlers/target/test-classes
INFO - Beginning load: /Users/dblevins/work/all/trunk/openejb/examples/webservice-
handlers/target/classes
INFO - Configuring enterprise application:
/Users/dblevins/work/all/trunk/openejb/examples/webservice-handlers
INFO - Auto-deploying ejb Calculator: EjbDeployment(deployment-id=Calculator)
INFO - Configuring Service(id=Default Singleton Container, type=Container, provider-
id=Default Singleton Container)
INFO - Auto-creating a container for bean Calculator: Container(type=SINGLETON,
id=Default Singleton Container)
INFO - Creating Container(id=Default Singleton Container)
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean org.superbiz.calculator.wsh.CalculatorTest:
Container(type=MANAGED, id=Default Managed Container)
INFO - Creating Container(id=Default Managed Container)
INFO - Enterprise application
"/Users/dblevins/work/all/trunk/openejb/examples/webservice-handlers" loaded.
INFO - Assembling app: /Users/dblevins/work/all/trunk/openejb/examples/webservice-
handlers
INFO - Created Ejb(deployment-id=Calculator, ejb-name=Calculator, container=Default
Singleton Container)
INFO - Started Ejb(deployment-id=Calculator, ejb-name=Calculator, container=Default
Singleton Container)
INFO - Deployed
Application(path=/Users/dblevins/work/all/trunk/openejb/examples/webservice-handlers)
INFO - Initializing network services
INFO - Creating ServerService(id=httpejbd)
INFO - Creating ServerService(id=cxf)
INFO - Creating ServerService(id=admin)
INFO - Creating ServerService(id=ejbd)
INFO - Creating ServerService(id=ejbds)
INFO - Initializing network services
INFO -   ** Starting Services **
INFO -   NAME                 IP              PORT
INFO -   httpejbd             127.0.0.1       4204
INFO - Creating Service {http://superbiz.org/wsdl}CalculatorService from class
org.superbiz.calculator.wsh.CalculatorWs
INFO - Setting the server's publish address to be http://nopath:80
INFO - Webservice(wsdl=http://127.0.0.1:4204/Calculator,
qname={http://superbiz.org/wsdl}CalculatorService) --> Ejb(id=Calculator)
INFO -   admin thread         127.0.0.1       4200
INFO -   ejbd                 127.0.0.1       4201
INFO -   ejbd                 127.0.0.1       4203
```

```
INFO - -------
INFO - Ready!
INFO - Creating Service {http://superbiz.org/wsdl}CalculatorService from WSDL:
http://127.0.0.1:4204/Calculator?wsdl
INFO - Creating Service {http://superbiz.org/wsdl}CalculatorService from WSDL:
http://127.0.0.1:4204/Calculator?wsdl
INFO - Default SAAJ universe not set
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.783 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
```

Inspecting the messages

The above would generate the following messages.

Calculator wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  name="CalculatorService" targetNamespace="http://superbiz.org/wsdl"
                  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
                  xmlns:tns="http://superbiz.org/wsdl" xmlns:xsd=
"http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xsd:schema attributeFormDefault="unqualified" elementFormDefault="unqualified"
                targetNamespace="http://superbiz.org/wsdl" xmlns:tns=
"http://superbiz.org/wsdl"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="multiply" type="tns:multiply"/>
      <xsd:complexType name="multiply">
        <xsd:sequence>
          <xsd:element name="arg0" type="xsd:int"/>
          <xsd:element name="arg1" type="xsd:int"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:element name="multiplyResponse" type="tns:multiplyResponse"/>
      <xsd:complexType name="multiplyResponse">
        <xsd:sequence>
          <xsd:element name="return" type="xsd:int"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:element name="sum" type="tns:sum"/>
      <xsd:complexType name="sum">
        <xsd:sequence>
          <xsd:element name="arg0" type="xsd:int"/>
          <xsd:element name="arg1" type="xsd:int"/>
```

```
      </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="sumResponse" type="tns:sumResponse"/>
    <xsd:complexType name="sumResponse">
      <xsd:sequence>
        <xsd:element name="return" type="xsd:int"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:schema>
</wsdl:types>
<wsdl:message name="multiplyResponse">
  <wsdl:part element="tns:multiplyResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="sumResponse">
  <wsdl:part element="tns:sumResponse" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="sum">
  <wsdl:part element="tns:sum" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="multiply">
  <wsdl:part element="tns:multiply" name="parameters">
  </wsdl:part>
</wsdl:message>
<wsdl:portType name="CalculatorWs">
  <wsdl:operation name="multiply">
    <wsdl:input message="tns:multiply" name="multiply">
    </wsdl:input>
    <wsdl:output message="tns:multiplyResponse" name="multiplyResponse">
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="sum">
    <wsdl:input message="tns:sum" name="sum">
    </wsdl:input>
    <wsdl:output message="tns:sumResponse" name="sumResponse">
    </wsdl:output>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="CalculatorServiceSoapBinding" type="tns:CalculatorWs">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="multiply">
    <soap:operation soapAction="" style="document"/>
    <wsdl:input name="multiply">
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="multiplyResponse">
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
```

```
    <wsdl:operation name="sum">
      <soap:operation soapAction="" style="document"/>
      <wsdl:input name="sum">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="sumResponse">
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="CalculatorService">
    <wsdl:port binding="tns:CalculatorServiceSoapBinding" name="CalculatorPort">
      <soap:address location="http://127.0.0.1:4204/Calculator?wsdl"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

SOAP sum and sumResponse

Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:sum xmlns:ns1="http://superbiz.org/wsdl">
      <arg0>4</arg0>
      <arg1>6</arg1>
    </ns1:sum>
  </soap:Body>
</soap:Envelope>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:sumResponse xmlns:ns1="http://superbiz.org/wsdl">
      <return>10001</return>
    </ns1:sumResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP multiply and multiplyResponse

Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:multiply xmlns:ns1="http://superbiz.org/wsdl">
      <arg0>3</arg0>
      <arg1>4</arg1>
    </ns1:multiply>
  </soap:Body>
</soap:Envelope>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:multiplyResponse xmlns:ns1="http://superbiz.org/wsdl">
      <return>12001</return>
    </ns1:multiplyResponse>
  </soap:Body>
</soap:Envelope>
```