

How to deploy a EJB as a Web Service on Axis 1.0

John Mammen

I will use a Rates Bean EJB to explain this example. Basically, this bean obtains rate details of a currency when an three digit currency code is passed.

I have four Classes. Rates.java, the remote bean, RatesBean.java, implementation bean, RatesHome.java home and RatesDetails return class for rate details. I will show excerpts of some these beans and deployment descriptor.(My J2EE application server is weblogic7.0)

Rates.java (Remote)

```
.....
public interface Rates extends EJBObject {

    public RatesDetails getRateDetails(String currencyCode) throws
    RemoteException,RateException;

    ...
}
```

RateDetails.java (JavaBean Return class)

```
...
public class RatesDetails implements Serializable{

    private String isoCode;
    private String vaildFrom;
    private String vaildTo;

    /** Holds value of property retailBuy. */
    private double retailBuy;

    /** Holds value of property retailMid. */
    private double retailMid;

    /** Holds value of property retailSell. */
    private double retailSell;

    /** Holds value of property wholesaleBuy. */
    private double wholesaleBuy;

    /** Holds value of property wholesaleMid. */
    private double wholesaleMid;

    /** Holds value of property wholesaleSell. */
    private double wholesaleSell;

    /** Holds value of property rateType. */
    private int rateType;

    public RatesDetails() {
        System.out.println("instantiating RatesDetails ");
    }

    public void setIsoCode(String isoCode){
        this.isoCode = isoCode;
    }
}
```

```

    public void setValidFrom(String vaildFrom){
        this.vaildFrom = vaildFrom;
    }

    public void setValidTo(String vaildTo){
        this.vaildTo = vaildTo;
    }

    //getter

    public String getIsoCode(){
        return isoCode;
    }

    public String getValidFrom(){
        return vaildFrom;
    }

    public String getValidTo(){
        return vaildTo;
    }

    /** Getter for property retailBuy.
     * @return Value of property retailBuy.
     */
    public double getRetailBuy() {
        return retailBuy;
    }

    /** Setter for property retailBuy.
     * @param retailBuy New value of property retailBuy.
     */
    public void setRetailBuy(double retailBuy) {
        this.retailBuy = retailBuy;
    }

    /** Getter for property retailMid.
     * @return Value of property retailMid.
     */
    public double getRetailMid() {
        return retailMid;
    }

    /** Setter for property retailMid.
     * @param retailMid New value of property retailMid.
     */
    public void setRetailMid(double retailMid) {
        this.retailMid = retailMid;
    }

    .....

```

ejb-jar.xml

```
<session>
  <ejb-name>RatesBean</ejb-name>
  <home>RatesHome</home>
  <remote>Rates</remote>
  <ejb-class>RatesBean</ejb-class>
  <session-type>Stateless</session-type>
  <transaction-type>Container</transaction-type>
  <resource-ref>
    <description><![CDATA[Datasource for New Rates
DB]]></description>
    <res-ref-name>jdbc/NewRatesDB</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
  </resource-ref>
</session>
```

```
<weblogic-enterprise-bean>
  <ejb-name>RatesBean</ejb-name>
  <stateless-session-descriptor>
    <pool>
      <max-beans-in-free-pool>50</max-beans-in-free-pool>
      <initial-beans-in-free-pool>5</initial-beans-in-free-pool>
    </pool>
  </stateless-session-descriptor>
  <reference-descriptor>
    <resource-description>
      <res-ref-name>jdbc/NewRatesDB</res-ref-name>
      <jndi-name>jdbc.RatePool</jndi-name>
    </resource-description>
  </reference-descriptor>
  <jndi-name>jndi.Rates</jndi-name> </weblogic-enterprise-
bean>
```

1. First make sure you have successfully deployed your EJB to your EJB Container.
2. Then install and configure Axis 1.0 with your J2EE application server.
3. Test the axis deployment by clicking on the validate of axis installation page.
4. Create a rates.wsdd in notepad as shown below. The below example corresponds to the above beans described. Please make note of the bolded text. Also make sure your classnames should include the packages if in packages. For example: if your **RatesHome** was in package com.mycompany, then your home interface name is **com.mycompany. RatesHome**

```
<?xml version="1.0" encoding="UTF-8"?>
<deployment xmlns="http://xml.apache.org/axis/wsdd/"
  xmlns:java="http://xml.apache.org/axis/wsdd/providers/java"
  xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance">
  <service name="RateDetails" provider="java:EJB">
    <parameter name="beanJndiName" value="jndi.Rates" />
    <parameter name="homeInterfaceName" value="RatesHome" />
    <parameter name="remoteInterfaceName" value="Rates" />
  </service>
</deployment>
```

```

        <parameter name="allowedMethods" value="getRateDetails" />
        <parameter name="className" value="RatesBean" />
        <parameter name="jndiURL" value="t3://localhost:7001" />
        <parameter name="jndiContextClass"
value="weblogic.jndi.WLInitialContextFactory" />
    <typeMapping
        xmlns:ns="http://soapinterop.org/xsd"
        qname="ns:RatesDetails"
        type="java:RatesDetails"
        serializer="org.apache.axis.encoding.ser.BeanSerializerFactory"
        deserializer="org.apache.axis.encoding.ser.BeanDeserializerFactory"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    />
</service>
</deployment>

```

5. Now deploy your rates.wsdd by issuing the command:

```

java org.apache.axis.client.AdminClient -
lhttp://localhost:7001/axis/services/ejbsservice rates.wsdd

```

6. If successfully deployed, you will not get any error message. Also take a look at your container console for any errors.

7. Now test your web service in the browser url:

<http://localhost:7001/axis/services/RateDetails?method=getRateDetails&value=USD>

If everything went ok, you should get something like this.

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <geRateDetailsResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
            <geRateDetailsReturn href="#id0"/>
        </geRateDetailsResponse>
        <multiRef id="id0" soapenc:root="0"
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xsi:type="ns1:RatesDetails"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://soapinterop.org/xsd">
            <rateType xsi:type="xsd:int">0</rateType>
            <wholesaleSell xsi:type="xsd:double">1.0</wholesaleSell>
            <retailSell xsi:type="xsd:double">1.0</retailSell>
            <retailMid xsi:type="xsd:double">1.0</retailMid>
            <wholesaleMid xsi:type="xsd:double">1.0</wholesaleMid>
            <validTo xsi:type="xsd:string" xsi:nil="true"/>
            <validFrom xsi:type="xsd:string" xsi:nil="true"/>
            <isoCode xsi:type="xsd:string" xsi:nil="true"/>
            <retailBuy xsi:type="xsd:double">1.0</retailBuy>
            <wholesaleBuy xsi:type="xsd:double">1.0</wholesaleBuy>
        </multiRef>
    </soapenv:Body>
</soapenv:Envelope>

```

```
</multiRef>  
</soapenv:Body>  
</soapenv:Envelope>
```

Hope this helps, if you have any problems with this post to axis-user@xml.apache.org