Jboss WS-AT example

I heard about WS-AT when I started using WebSphere Application Server, but never tried it. This time I got a chance to really try this out with Jboss 7. So, I have come up with a real time example which uses Spring 3, Hibernate 3 and MySQL 5.1.

## Pre-requisites

1. SpringSource Tool Suite (Version: 2.9.1.RELEASE) or Maven integrated Eclipse IDE.
2. Jboss Application Server 7.1.1
   * <http://download.jboss.org/jbossas/7.1/jboss-as-7.1.1.Final/jboss-as-7.1.1.Final.zip>
3. MySQL 5.1
   * <http://dev.mysql.com/downloads/mysql/5.1.html>
     + Download **Windows (x86, 64-bit), MSI Installer**
4. MySQL Connector/J
   * <http://dev.mysql.com/downloads/connector/j/>
     + Download mysql-connector-java-5.1.22.zip
5. JAX-WS RI tool
   * <http://jax-ws.java.net/2.1.1/>
     + Extract the zip and install jax ws ri

## Setup

This section describes on configuring XTS, Datasources, Spring 3 and MySQL libraries in Jboss AS 7.

* + Extract **jboss-as-7.1.1.Final.zip** to a desired location (example : C: \servers)

Spring libraries

* + Create “**org\springframework\spring\main**“ folder under **modules**
  + Copy below jars into newly created **main** folder
    - **cglib-nodep-2.2.2.jar**
    - **aopalliance-1.0.jar**
    - **spring-aop-3.1.1.RELEASE.jar**
    - **spring-asm-3.1.1.RELEASE.jar**
    - **spring-beans-3.1.1.RELEASE.jar**
    - **spring-context-3.1.1.RELEASE.jar**
    - **spring-context-support-3.1.1.RELEASE.jar**
    - **spring-core-3.1.1.RELEASE.jar**
    - **spring-expression-3.1.1.RELEASE.jar**
    - **spring-jdbc-3.1.1.RELEASE.jar**
    - **spring-orm-3.1.1.RELEASE.jar**
    - **spring-tx-3.1.1.RELEASE.jar**
    - **spring-webmvc-3.1.1.RELEASE.jar**
    - **spring-web-3.1.1.RELEASE.jar**
  + Create a **module.xml** under this location and copy the below xml instance.

**<?xml version="1.0" encoding="UTF-8" ?>**

**<module xmlns="urn:jboss:module:1.0" name="org.springframework.spring">**

**<resources>**

**<resource-root path="cglib-nodep-2.2.2.jar" />**

**<resource-root path="aopalliance-1.0.jar" />**

**<resource-root path="spring-aop-3.1.1.RELEASE.jar" />**

**<resource-root path="spring-asm-3.1.1.RELEASE.jar" />**

**<resource-root path="spring-beans-3.1.1.RELEASE.jar" />**

**<resource-root path="spring-context-3.1.1.RELEASE.jar" />**

**<resource-root path="spring-context-support-3.1.1.RELEASE.jar" />**

**<resource-root path="spring-core-3.1.1.RELEASE.jar" />**

**<resource-root path="spring-expression-3.1.1.RELEASE.jar" />**

**<resource-root path="spring-jdbc-3.1.1.RELEASE.jar" />**

**<resource-root path="spring-orm-3.1.1.RELEASE.jar" />**

**<resource-root path="spring-tx-3.1.1.RELEASE.jar" />**

**<resource-root path="spring-webmvc-3.1.1.RELEASE.jar" />**

**<resource-root path="spring-web-3.1.1.RELEASE.jar" />**

**</resources>**

**<dependencies>**

**<module name="org.apache.commons.logging" />**

**<module name="org.slf4j" />**

**<module name="javax.servlet.api" />**

**<module name="javax.faces.api" slot="1.2" />**

**<module name="com.sun.jsf-impl" slot="1.2" />**

**<module name="org.jboss.vfs" />**

**<module name="javax.persistence.api" />**

**<module name="javax.xml.bind.api" />**

**<module name="javax.api" />**

**<module name="javax.jms.api" />**

**<module name="javax.annotation.api" />**

**<module name="org.apache.commons.logging" />**

**<module name="org.hibernate" />**

**</dependencies>**

**</module>**

MySQL jdbc driver

* Create folder “**com\mysql\jdbc\main**” under **modules**
* Copy **mysql-connector-java-5.1.XX-bin.jar** to this newly created main folder
* Create a **module.xml** and copy the below xml instance.

**<?xml version="1.0" encoding="UTF-8"?>**

**<module xmlns="urn:jboss:module:1.0" name="com.mysql.jdbc">**

**<resources>**

**<resource-root path="mysql-connector-java-5.1.XX-bin.jar"/>**

**</resources>**

**<dependencies>**

**<module name="javax.api"/>**

**<module name="javax.transaction.api"/>**

**</dependencies>**

**</module>**

DataSources

Install MySQL on your machine and configure a DB instance with default settings. Make sure that instance is up and running.

We will create two MySQL datasources

“**java:jboss/datasources/MySqlDS**” – Used by Web service Server application

“**java:jboss/datasources/MySqlXADS**” – Used by Web service Client application.

In order to configure these two data sources add following xml snippet into “**standalone.xml**” available under

${JBOSS\_ROOT}/ standalone/configuration

Search for -

**<subsystem xmlns="urn:jboss:domain:datasources:1.0">**

Under <datasources> node, add the following

**<datasource jndi-name="java:jboss/datasources/MySqlDS" pool-name="MySqlDS" enabled="true" use-java-context="true">**

**<connection-url>jdbc:mysql://{HOST}/{DB\_NAME}</connection-url>**

**<driver>mysql</driver>**

**<security>**

**<user-name>#USER\_NAME </user-name>**

**<password>#PASSWORD </password>**

**</security>**

**</datasource>**

**<xa-datasource jndi-name="java:jboss/datasources/MySqlXADS" pool-name="MySqlXADS" enabled="true" use-java-context="true">**

**<xa-datasource-property name="ServerName"> localhost</xa-datasource-property>**

**<xa-datasource-property name="DatabaseName">#DB\_NAME</xa-datasource-property>**

**<xa-datasource-property name="User">#USER\_NAME</xa-datasource-property>**

**<xa-datasource-property name="Password">#PASSWORD</xa-datasource-property>**

**<driver>mysql</driver>**

**</xa-datasource>**

Under <drivers> node, add following

**<driver name="mysql" module="com.mysql.jdbc">**

**<xa-datasource-class>com.mysql.jdbc.jdbc2.optional.MysqlXADataSource</xa-datasource-class>**

**</driver>**

XTS

Jboss by default doesn’t have XTS enabled. In order get this example working, it is necessary that we enable the XTS extension by adding its related configuration in **standalone.xml**

Under <extensions> add the following

**<extension module="org.jboss.as.xts"/>**

Search for

**<subsystem xmlns="urn:jboss:domain:transactions:1.1">**

Add the following

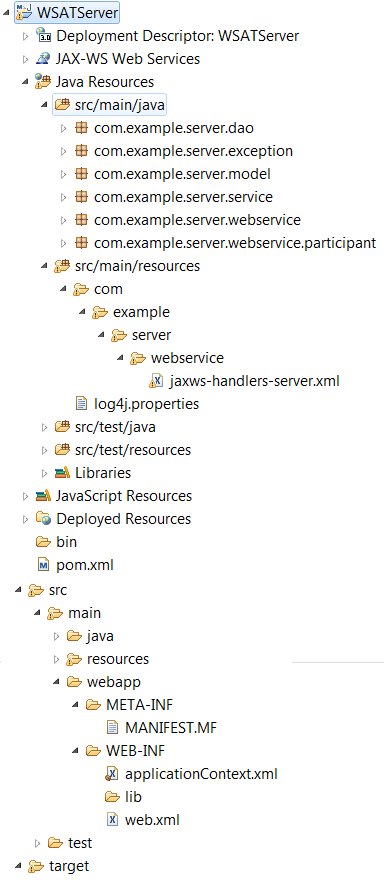
**<subsystem xmlns="urn:jboss:domain:xts:1.0">**

**<xts-environment url="http://localhost:8080/ws-c11/ActivationService"/>**

**</subsystem>**

# WS-AT Server Project

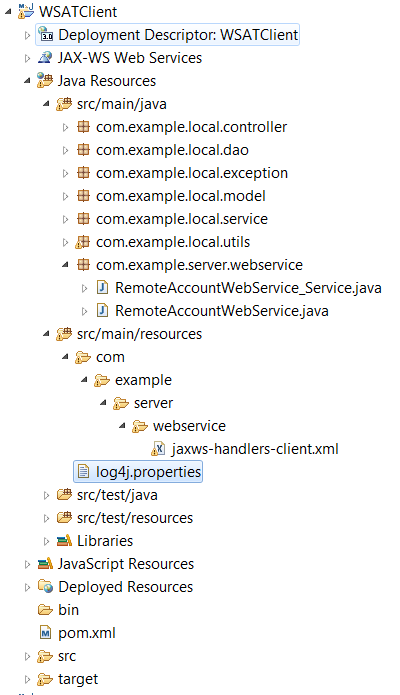
This is a JAX-WS webservice application which coordinates with calling application using Transaction Bridge.



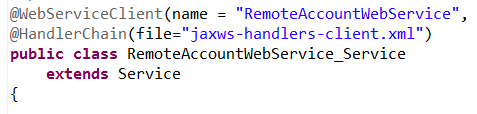
The JbossTS Bridge for the service is registered in an xml file which consists of Inbound Transaction Bridge Handler and JaxWSHeaderContextProcessor. This xml file is made available to the webservice via @HandlerChain annotation. Once the service is configured to use the incoming transaction context, the web service can then make a call to any business logic which requires transaction. Point to note here is that the WS-AT handlers always expect the JTA transaction context to be available when the webservice is invoked. For better understanding of Transaction Bridge one can refer to <http://docs.jboss.org/jbosstm/4.17.1.Final/guides/txbridge_guide/index.html>

# WS-AT Client Project

This is a restful webservice which invokes WS AT enabled remote webservice. This consists of JAXWS -RI generated remote service stubs which will be used for invocation.



The remote service artifacts have been configured to propagate the transactions to remote webservice via an Outbound Transaction Bridge similar to the Inbound Transaction Bridge used on the remote service side.



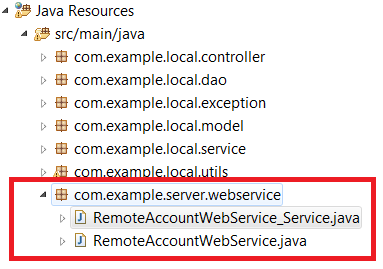
## JAXWS-RI: Generate WS Client artifacts

This section describes on generating WS client artifacts.

C:\>SET PATH=%PATH%;C:\JAXWS\_RI\bin

C:\>wsimport -s C:\Temp\WSATServiceStubs <http://localhost:8080/WSATServer/RemoteAccountWebService?wsdl>

The above command generates the WS client class and an interface. Import these into ws client project.



# Test Cases

Test case 1: <http://localhost:8080/WSATClient/saveAccountRecord.action?accountNumber=AMX1212312&amount=19219&testCaseId=SUCCESS>

Test case 2:

<http://localhost:8080/WSATClient/saveAccountRecord.action?accountNumber=AMX1212312&amount=19219&testCaseId=LOCAL_ROLLBACK>

Test case 3:

<http://localhost:8080/WSATClient/saveAccountRecord.action?accountNumber=AMX1212312&amount=19219&testCaseId=REMOTE_ROLLBACK>

# References

<http://docs.jboss.org/jbosstm/4.17.1.Final/guides/txbridge_guide/index.html>

<http://specs.xmlsoap.org/ws/2004/10/wsat/wsat.pdf>

<http://docs.jboss.org/jbosstm/docs/4.2.3/manuals/pdf/xts/ProgrammersGuide.pdf>