Simple MDB with Descriptor

Example simple-mdb-with-descriptor can be browsed at https://github.com/apache/tomee/tree/master/examples/simple-mdb-with-descriptor

Help us document this example! Click the blue pencil icon in the upper right to edit this page.

ChatBean

```
package org.superbiz.mdbdesc;

import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.DeliveryMode;
import javax.jms.JMSException;
import javax.jms.Message;
import javax.jms.MessageListener;
import javax.jms.MessageProducer;
import javax.jms.Queue;
import javax.jms.Session;
import javax.jms.TextMessage;

public class ChatBean implements MessageListener {

    private ConnectionFactory connectionFactory;

    private Queue answerQueue;

    public void onMessage(Message message) {
        try {

            final TextMessage textMessage = (TextMessage) message;
            final String question = textMessage.getText();

            if ("Hello World!".equals(question)) {

                respond("Hello, Test Case!");
            } else if ("How are you?".equals(question)) {

                respond("I'm doing well.");
            } else if ("Still spinning?".equals(question)) {

                respond("Once every day, as usual.");
            }
        } catch (JMSException e) {
            throw new IllegalStateException(e);
        }
    }
    private void respond(String text) throws JMSException {

        Connection connection = null;
        Session session = null;

        try {
            connection = connectionFactory.createConnection();
            connection.start();

            // Create a Session
            session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);

            // Create a MessageProducer from the Session to the Topic or Queue
            MessageProducer producer = session.createProducer(answerQueue);
            producer.setDeliveryMode(DeliveryMode.NON_PERSISTENT);

            // Create a message
            TextMessage message = session.createTextMessage(text);

            // Tell the producer to send the message
            producer.send(message);
        } finally {
            // Clean up
            if (session != null) session.close();
            if (connection != null) connection.close();
        }
    }
}
```

ejb-jar.xml

```
<ejb-jar xmlns="http://java.sun.com/xml/ns/javaee" metadata-complete="true">
  <enterprise-beans>

    <message-driven>

      <ejb-name>ChatBean</ejb-name>
      <ejb-class>org.superbiz.mdbdesc.ChatBean</ejb-class>

      <messaging-type>javax.jms.MessageListener</messaging-type>

      <activation-config>
        <activation-config-property>
          <activation-config-property-name>destination</activation-config-property-
name>
          <activation-config-property-value>ChatBean</activation-config-property-
value>
        </activation-config-property>
        <activation-config-property>
          <activation-config-property-name>destinationType</activation-config-
property-name>
          <activation-config-property-value>javax.jms.Queue</activation-config-
property-value>
        </activation-config-property>
      </activation-config>

      <resource-ref>
        <res-ref-name>
java:comp/env/org.superbiz.mdbdesc.ChatBean/connectionFactory</res-ref-name>
        <res-type>javax.jms.ConnectionFactory</res-type>
        <injection-target>
          <injection-target-class>org.superbiz.mdbdesc.ChatBean</injection-target-
class>
          <injection-target-name>connectionFactory</injection-target-name>
        </injection-target>
      </resource-ref>

      <resource-env-ref>
        <resource-env-ref-name>java:comp/env/AnswerQueue</resource-env-ref-name>
        <resource-env-ref-type>javax.jms.Queue</resource-env-ref-type>
        <mapped-name>AnswerQueue</mapped-name>
        <injection-target>
          <injection-target-class>org.superbiz.mdbdesc.ChatBean</injection-target-
class>
          <injection-target-name>answerQueue</injection-target-name>
        </injection-target>
      </resource-env-ref>

    </message-driven>

  </enterprise-beans>
</ejb-jar>
```

ChatBeanTest

```
package org.superbiz.mdb;

import junit.framework.TestCase;

import javax.annotation.Resource;
import javax.ejb.embeddable.EJBContainer;
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.JMSException;
import javax.jms.MessageConsumer;
import javax.jms.MessageProducer;
import javax.jms.Queue;
import javax.jms.Session;
import javax.jms.TextMessage;

public class ChatBeanTest extends TestCase {

    @Resource
    private ConnectionFactory connectionFactory;

    @Resource(name = "ChatBean")
    private Queue questionQueue;

    @Resource(name = "AnswerQueue")
    private Queue answerQueue;

    public void test() throws Exception {

        EJBContainer.createEJBContainer().getContext().bind("inject", this);

        final Connection connection = connectionFactory.createConnection();

        connection.start();

        final Session session = connection.createSession(false, Session
.AUTO_ACKNOWLEDGE);

        final MessageProducer questions = session.createProducer(questionQueue);

        final MessageConsumer answers = session.createConsumer(answerQueue);


        sendText("Hello World!", questions, session);

        assertEquals("Hello, Test Case!", receiveText(answers));


        sendText("How are you?", questions, session);

        assertEquals("I'm doing well.", receiveText(answers));


        sendText("Still spinning?", questions, session);

        assertEquals("Once every day, as usual.", receiveText(answers));
    }

    private void sendText(String text, MessageProducer questions, Session session)
throws JMSException {

        questions.send(session.createTextMessage(text));
    }

```

```java
    private String receiveText(MessageConsumer answers) throws JMSException {

        return ((TextMessage) answers.receive(1000)).getText();
    }
}
```

Running

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running org.superbiz.mdb.ChatBeanTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/simple-mdb-with-descriptor
INFO - openejb.base = /Users/dblevins/examples/simple-mdb-with-descriptor
INFO - Using 'javax.ejb.embeddable.EJBContainer=true'
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 - Found EjbModule in classpath: /Users/dblevins/examples/simple-mdb-with-descriptor/target/classes
INFO - Beginning load: /Users/dblevins/examples/simple-mdb-with-descriptor/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/simple-mdb-with-descriptor
WARN - Method 'lookup' is not available for 'javax.annotation.Resource'. Probably using an older Runtime.
INFO - Configuring Service(id=Default MDB Container, type=Container, provider-id=Default MDB Container)
INFO - Auto-creating a container for bean ChatBean: Container(type=MESSAGE, id=Default MDB Container)
INFO - Configuring Service(id=Default JMS Resource Adapter, type=Resource, provider-id=Default JMS Resource Adapter)
INFO - Configuring Service(id=Default JMS Connection Factory, type=Resource, provider-id=Default JMS Connection Factory)
INFO - Auto-creating a Resource with id 'Default JMS Connection Factory' of type 'javax.jms.ConnectionFactory for 'ChatBean'.
INFO - Auto-linking resource-ref 'java:comp/env/org.superbiz.mdbdesc.ChatBean/connectionFactory' in bean ChatBean to Resource(id=Default JMS Connection Factory)
INFO - Configuring Service(id=AnswerQueue, type=Resource, provider-id=Default Queue)
INFO - Auto-creating a Resource with id 'AnswerQueue' of type 'javax.jms.Queue for 'ChatBean'.
INFO - Auto-linking resource-env-ref 'java:comp/env/AnswerQueue' in bean ChatBean to Resource(id=AnswerQueue)
INFO - Configuring Service(id=ChatBean, type=Resource, provider-id=Default Queue)
INFO - Auto-creating a Resource with id 'ChatBean' of type 'javax.jms.Queue for
'ChatBean'.
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean org.superbiz.mdb.ChatBeanTest:
Container(type=MANAGED, id=Default Managed Container)
INFO - Auto-linking resource-ref
'java:comp/env/org.superbiz.mdb.ChatBeanTest/connectionFactory' in bean
org.superbiz.mdb.ChatBeanTest to Resource(id=Default JMS Connection Factory)
INFO - Auto-linking resource-env-ref 'java:comp/env/AnswerQueue' in bean
org.superbiz.mdb.ChatBeanTest to Resource(id=AnswerQueue)
INFO - Auto-linking resource-env-ref 'java:comp/env/ChatBean' in bean
org.superbiz.mdb.ChatBeanTest to Resource(id=ChatBean)
INFO - Enterprise application "/Users/dblevins/examples/simple-mdb-with-descriptor"
loaded.
INFO - Assembling app: /Users/dblevins/examples/simple-mdb-with-descriptor
INFO -
Jndi(name="java:global/EjbModule1842275169/org.superbiz.mdb.ChatBeanTest!org.superbiz.
mdb.ChatBeanTest")
INFO - Jndi(name="java:global/EjbModule1842275169/org.superbiz.mdb.ChatBeanTest")
INFO - Created Ejb(deployment-id=org.superbiz.mdb.ChatBeanTest, ejb-
name=org.superbiz.mdb.ChatBeanTest, container=Default Managed Container)
INFO - Created Ejb(deployment-id=ChatBean, ejb-name=ChatBean, container=Default MDB
Container)
INFO - Started Ejb(deployment-id=org.superbiz.mdb.ChatBeanTest, ejb-
name=org.superbiz.mdb.ChatBeanTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=ChatBean, ejb-name=ChatBean, container=Default MDB
Container)
INFO - Deployed Application(path=/Users/dblevins/examples/simple-mdb-with-descriptor)
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.914 sec

Results :

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