Simple MDB

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

Below is a fun app, a chat application that uses JMS. We create a message driven bean, by marking our class with `@MessageDriven`. A message driven bean has some similarities with a stateless session bean, in the part that it is pooled too.

Well, lets tell our chat-app to listen for incoming messages. That we do by implementing `MessageListener` and overriding the `onMessage(Message message)`.

Then this app "listens" for incoming messages, and the messages picked up are processed by `onMessage(Message message)` method.

That finishes our message driven bean implementation. The "processing" part could be anything that fits your business-requirement.

In this case, it is to respond to the user. The `respond` method shows how a Message can be sent.

This sequence diagram shows how a message is sent.

ChatBean

```java
package org.superbiz.mdb;

import javax.annotation.Resource;
import javax.ejb.MessageDriven;
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;

@MessageDriven
public class ChatBean implements MessageListener {

    @Resource
    private ConnectionFactory connectionFactory;
```

```java
    @Resource(name = "AnswerQueue")
    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();
        }
    }
```

```
}
```

ChatBeanTest

```java
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));
    }

    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
INFO - openejb.base = /Users/dblevins/examples/simple-mdb
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/target/classes
INFO - Beginning load: /Users/dblevins/examples/simple-mdb/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/simple-mdb
WARN - Method 'lookup' is not available for 'javax.annotation.Resource'. Probably
using an older Runtime.
INFO - Auto-configuring a message driven bean ChatBean destination ChatBean to be
destinationType javax.jms.Queue
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.mdb.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" loaded.
INFO - Assembling app: /Users/dblevins/examples/simple-mdb
INFO -
Jndi(name="java:global/EjbModule1515710343/org.superbiz.mdb.ChatBeanTest!org.superbiz.
mdb.ChatBeanTest")
INFO - Jndi(name="java:global/EjbModule1515710343/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)
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.547 sec

Results :

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