Simple MDB and CDI

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

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

ChatBean

```
package org.superbiz.mdb;

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

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

    @Inject
    private ChatRespondCreator responder;

    public void onMessage(Message message) {
        try {

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

            if (response != null) {
                respond(response);
            }
        } 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();
        }
    }
}
```

ChatRespondCreator

```
package org.superbiz.mdb;

public class ChatRespondCreator {
    public String respond(String question) {
        if ("Hello World!".equals(question)) {
            return "Hello, Test Case!";
        } else if ("How are you?".equals(question)) {
            return "I'm doing well.";
        } else if ("Still spinning?".equals(question)) {
            return "Once every day, as usual.";
        }
        return null;
    }
}
```

beans.xml

```
<!--

    Licensed to the Apache Software Foundation (ASF) under one or more
    contributor license agreements.  See the NOTICE file distributed with
    this work for additional information regarding copyright ownership.
    The ASF licenses this file to You under the Apache License, Version 2.0
    (the "License"); you may not use this file except in compliance with
    the License.  You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0
    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
-->
<beans/>
```

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));
    }

    private String receiveText(MessageConsumer answers) throws JMSException {

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