|  |  |
| --- | --- |
| package ejb30; | |
| 02 | import javax.ejb.MessageDriven; | |

|  |  |  |
| --- | --- | --- |
| 03 | import javax.jms.JMSException; | |
| 04 | import javax.jms.Message; |

|  |  |  |
| --- | --- | --- |
| 05 | import javax.jms.MessageListener; | |
| 06 | import javax.jms.TextMessage; |

|  |  |  |
| --- | --- | --- |
| 07 | import javax.ejb.ActivationConfigProperty; | |
| 08 |  |

|  |  |
| --- | --- |
| 09 | @MessageDriven( |
| 10 | activationConfig = { | |

|  |  |  |
| --- | --- | --- |
| 11 | | @ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue") , |
| 12 | @ActivationConfigProperty(propertyName="connectionFactoryJndiName",propertyValue="QCF"), | |

|  |  |  |
| --- | --- | --- |
| 13 | @ActivationConfigProperty(propertyName="destinationJndiName", propertyValue="TestQ") | |
| 14 | } |

|  |  |  |
| --- | --- | --- |
| 15 | ,mappedName="TestQ" | |
| 16 | ) |

|  |  |
| --- | --- |
| 17 |  |
| 18 | public class MyMDB implements MessageListener | |

|  |  |
| --- | --- |
| 19 | { |
| 20 | public void onMessage(Message message) | |

|  |  |
| --- | --- |
| 21 | { |
| 22 | TextMessage textMessage = (TextMessage) message; | |

|  |  |
| --- | --- |
| 23 | try { |
| 24 | System.out.println("\n\n\t(mdb) MyMDB Received \n"+ textMessage.getText()); | |

|  |  |
| --- | --- |
| 25 | } |
| 26 | catch (JMSException e) | |

|  |  |
| --- | --- |
| 27 | { |
| 28 | e.printStackTrace(); | |

|  |  |
| --- | --- |
| 29 | } |
| 30 | } |

|  |  |
| --- | --- |
| 31 | } |

**Step 2). Compile the Above MDB class and then create a JAR out of it.**

javac    –d    .   MyMDB.java

jar   -cvf     mdb30.jar   ejb30     MyMDB.java

**Step 3). Now Logic to WebLogic Admin console and create a ConnectionFactory with JNDI Name “QCF” and a Queue with JNDI Name “TestQ”.**

**Step 4). Deploy the MDB Jar (mdb30) file on the Server.**

**Step 5). Now start Writing the Client Queue Sender Program which is going to send some JMS messages to the TestQ and as a listener the MDB is going to consume these JMS messages.**

|  |  |
| --- | --- |
| 01 | import java.io.IOException; |
| 02 | import java.util.Hashtable; |

|  |  |
| --- | --- |
| 03 |  |
| 04 | import javax.jms.JMSException; | |

|  |  |
| --- | --- |
| 05 | import javax.jms.Queue; |
| 06 | import javax.jms.QueueConnection; | |

|  |  |  |
| --- | --- | --- |
| 07 | import javax.jms.QueueConnectionFactory; | |
| 08 | import javax.jms.QueueSender; |

|  |  |  |
| --- | --- | --- |
| 09 | import javax.jms.QueueSession; | |
| 10 | import javax.jms.Session; |

|  |  |  |
| --- | --- | --- |
| 11 | import javax.jms.TextMessage; | |
| 12 | import javax.naming.Context; |

|  |  |
| --- | --- |
| 13 | import javax.naming.InitialContext; |
| 14 | import javax.naming.NamingException; | |

|  |  |
| --- | --- |
| 15 |  |
| 16 | public class QueueSend | |

|  |  |
| --- | --- |
| 17 | { |
| 18 | // Defines the JNDI context factory. | |

|  |  |  |
| --- | --- | --- |
| 19 | public final static String JNDI\_FACTORY="weblogic.jndi.WLInitialContextFactory"; | |
| 20 |  |

|  |  |
| --- | --- |
| 21 | // Defines the JMS context factory. |
| 22 | public final static String JMS\_FACTORY="QCF"; | |

|  |  |
| --- | --- |
| 23 |  |
| 24 | // Defines the queue. | |

|  |  |  |
| --- | --- | --- |
| 25 | public final static String QUEUE="TestQ"; | |
| 26 |  |

|  |  |  |
| --- | --- | --- |
| 27 | private QueueConnectionFactory qconFactory; | |
| 28 | private QueueConnection qcon; |

|  |  |  |
| --- | --- | --- |
| 29 | private QueueSession qsession; | |
| 30 | private QueueSender qsender; |

|  |  |
| --- | --- |
| 31 | private Queue queue; |
| 32 | private TextMessage msg; | |

|  |  |
| --- | --- |
| 33 |  |
| 34 | public void init(Context ctx, String queueName)    throws NamingException, JMSException | |

|  |  |
| --- | --- |
| 35 | { |
| 36 | qconFactory = (QueueConnectionFactory) ctx.lookup(JMS\_FACTORY); | |

|  |  |
| --- | --- |
| 37 | qcon = qconFactory.createQueueConnection(); |
| 38 | qsession = qcon.createQueueSession(false, Session.AUTO\_ACKNOWLEDGE); | |

|  |  |
| --- | --- |
| 39 | queue = (Queue) ctx.lookup(queueName); |
| 40 | qsender = qsession.createSender(queue); | |

|  |  |  |
| --- | --- | --- |
| 41 | msg = qsession.createTextMessage(); | |
| 42 | qcon.start(); |

|  |  |  |
| --- | --- | --- |
| 43 | } | |
| 44 |  |

|  |  |  |
| --- | --- | --- |
| 45 | public void send(String message,int counter) throws JMSException { | |
| 46 | msg.setText(message); |

|  |  |  |
| --- | --- | --- |
| 47 | msg.setIntProperty("counter", counter); | |
| 48 | qsender.send(msg); |

|  |  |  |
| --- | --- | --- |
| 49 | } | |
| 50 |  |

|  |  |  |
| --- | --- | --- |
| 51 | public void close() throws JMSException { | |
| 52 | qsender.close(); |

|  |  |  |
| --- | --- | --- |
| 53 | qsession.close(); | |
| 54 | qcon.close(); |

|  |  |  |
| --- | --- | --- |
| 55 | } | |
| 56 |  |

|  |  |  |
| --- | --- | --- |
| 57 | public static void main(String[] args) throws Exception { | |
| 58 | if (args.length != 1) { |

|  |  |  |
| --- | --- | --- |
| 59 | System.out.println("Usage: java QueueSend WebLogicURL"); | |
| 60 | return; |

|  |  |
| --- | --- |
| 61 | } |
| 62 | InitialContext ic = getInitialContext(args[0]); | |

|  |  |  |
| --- | --- | --- |
| 63 | QueueSend qs = new QueueSend(); | |
| 64 | qs.init(ic, QUEUE); |

|  |  |  |
| --- | --- | --- |
| 65 | readAndSend(qs); | |
| 66 | qs.close(); |

|  |  |  |
| --- | --- | --- |
| 67 | } | |
| 68 |  |

|  |  |  |
| --- | --- | --- |
| 69 | private static void readAndSend(QueueSend qs)    throws IOException, JMSException | |
| 70 | { |

|  |  |  |
| --- | --- | --- |
| 71 | String line="Test Message Body with counter = "; | |
| 72 | for(int i=0;i<10;i++){ |

|  |  |
| --- | --- |
| 73 | qs.send(line+i,i); |
| 74 | System.out.println("JMS Message Sent: "+line+i+"\n"); | |

|  |  |
| --- | --- |
| 75 | } |
| 76 | } |

|  |  |
| --- | --- |
| 77 |  |
| 78 | private static InitialContext getInitialContext(String url)    throws NamingException | |

|  |  |
| --- | --- |
| 79 | { |
| 80 | Hashtable env = new Hashtable(); | |

|  |  |  |
| --- | --- | --- |
| 81 | env.put(Context.INITIAL\_CONTEXT\_FACTORY, JNDI\_FACTORY); | |
| 82 | env.put(Context.PROVIDER\_URL, url); |

|  |  |  |
| --- | --- | --- |
| 83 | return new InitialContext(env); | |
| 84 | } |

|  |  |
| --- | --- |
| 85 |  |
| 86 | } | |

**Step 6). Compile the above program and then run it.**

javac        –d     .       QueueSend.java

java    QueueSend   t3://localhost:7001

**Step 7). Now in Admin Console Go to the Monitoring Tab of the JMS Queue and see how many messages are getting consumed by MDB and how many messages are pending in the queue.**

- See more at: http://middlewaremagic.com/weblogic/?p=62#sthash.odCbjtNs.dpuf