Message Driven Bean (MDB) példa - Üzenet szűrés

Hozzunk létre egy olyan MDBMessageSelector osztályt, amely csak a “pirosnak jelölt” üzeneteket fogja fogadni az üzenetsorról.

```
  @MessageDriven(name = "MDBMessageSelectorExample",
                 activationConfig =
                       {
                          @ActivationConfigProperty(propertyName =
"destinationType", propertyValue = "javax.jms.Queue"),
                          @ActivationConfigProperty(propertyName =
"destination", propertyValue = "ColorQueue"),
                          @ActivationConfigProperty(propertyName =
"messageSelector", propertyValue = "color = 'RED'")
                       })
  @TransactionManagement(value= TransactionManagementType.CONTAINER)
  @TransactionAttribute(value= TransactionAttributeType.REQUIRED)

  public class MDBMessageSelector implements MessageListener
  {
   public void onMessage(Message message)
   {
      try
      {
         TextMessage textMessage = (TextMessage)message;
         String text = textMessage.getText();
         String color = textMessage.getStringProperty("color");
         System.out.println("message " + text + " received color=" + color);
      }
      catch (Exception e)
      {
         e.printStackTrace();
      }
   }
  }
```

A standalone-full.xml-ben adjuk hozzá a távoli elérést:

```
  <jms-queue name="ColorQueue" entries="ColorQueue
java:jboss/exported/jms/queue/ColorQueue"/>
```

A hozzá tartozó klienst viszont most ne a dinamikus web projektben, hanem egy külön java projektben hozzuk létre. A projektre jobb egérgombbal kattintva: Properties/Java Build Path/Add External Jar - segítségével adjuk hozzá a jboss-client.jar-t a [wildfly]\bin\client\ könyvtárból.

A [wildfly]\bin\add-user.bat segítségével egy távoli felhasználót is létre kell hozni. A példában a felhasználónév:quser és a jelszó:Password_1 és a “guest” csoportba kell tartoznia.

```
import java.util.Hashtable;

import javax.jms.Queue;
import javax.jms.QueueConnection;
import javax.jms.QueueConnectionFactory;
import javax.jms.QueueSender;
import javax.jms.QueueSession;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.naming.Context;
import javax.naming.NamingException;

import org.wildfly.naming.client.WildFlyInitialContextFactory;

public class TestRemoteQueue {

    public static void main(String[] args) throws Exception {
        Context initialContext = null;
        try {
            initialContext = getInitialContext();
            QueueConnectionFactory qconFactory = (QueueConnectionFactory)
initialContext
                    .lookup("jms/RemoteConnectionFactory");
            QueueConnection qcon =
qconFactory.createQueueConnection("quser", "Password_1");
            QueueSession qsession = qcon.createQueueSession(false,
Session.AUTO_ACKNOWLEDGE);
            Queue queue = (Queue)
initialContext.lookup("java:/jms/queue/TaskQueue");
            QueueSender qsender = qsession.createSender(queue);
            qcon.start();
            TextMessage msg = qsession.createTextMessage();
            msg.setStringProperty("color", "RED");
            msg.setText("Hello world! :)");
            qsender.send(msg);
            qsender.close();
            qsession.close();
            qcon.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static Context getInitialContext() throws NamingException {
        Hashtable env = new Hashtable();
        env.put(Context.INITIAL_CONTEXT_FACTORY,
"org.wildfly.naming.client.WildFlyInitialContextFactory");
        env.put(Context.PROVIDER_URL, "remote+http://127.0.0.1:8080");
        return new WildFlyInitialContextFactory().getInitialContext(env);
    }
```

```
}
```

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:informacios_rendszerek_integralasa:uezenet_szurese

Last update: **2022/05/06 17:15**