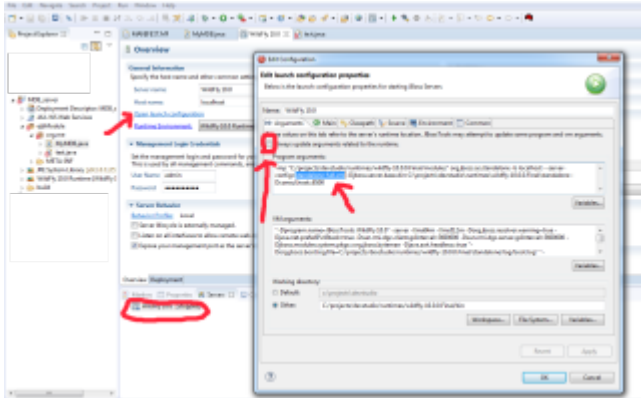


Pont-Pont példa

A JMS modul bekapcsolása a Wildfly alkalmazás szerverben. A JMS komponens, a standalone-full.xml konfigurációban szerepel, az alap konfiguráció (standalone.xml) nem tartalmazza. Az Jboss Dev. Studio "Launch configurations"-nál be kell állítani a "-server-config=standalone-full.xml" és újraindítani a wildfly-t. A pirossal jelzett checkboxot is ki kell kapcsolni.



Hozzunk létre egy projekt-et (MDB-test néven): dinamikus web vagy EJB projekt is lehet, és az alábbi osztályt adjuk meg:

```
package org.ait;

import java.util.Date;

import jakarta.ejb.ActivationConfigProperty;
import jakarta.ejb.MessageDriven;
import jakarta.jms.JMSException;
import jakarta.jms.Message;
import jakarta.jms.MessageListener;
import jakarta.jms.TextMessage;

@MessageDriven(name = "TaskQueue", activationConfig = {
    @ActivationConfigProperty(propertyName = "destinationLookup",
        propertyValue = "TaskQueue"),
    @ActivationConfigProperty(propertyName = "destinationType",
        propertyValue = "javax.jms.Queue"),
    @ActivationConfigProperty(propertyName = "acknowledgeMode",
        propertyValue = "Auto-acknowledge")})
public class TaskGenerator implements MessageListener {

    @Override
    public void onMessage(Message message) {
        try {
            if (message instanceof TextMessage) {
                System.out.println("Az üzenet ekkor megérkezett: " + new
Date());
                TextMessage msg = (TextMessage) message;
                System.out.println("Az üzenet: " + msg.getText());
            } else {
```

```
        System.out.println("hibás üzenet");
    }
    } catch (JMSEException e) {
        e.printStackTrace();
    }
}
}
```

Az üzenetsor TaskQueue néven automatikusan létrejön, az alábbi annotáció miatt:

```
@ActivationConfigProperty(propertyName = "destinationLookup", propertyValue = "TaskQueue")
```

Távoli elérés az üzenetsorhoz

Ahhoz, hogy távolról is el lehessen érni egy üzenetsort, a JNDI nevének "java:jboss/exported/" kell kezdődnie. Ehhez a standalone-full.xml-ben a messaging-activemq részhez tegyük be a <server name="default">-tag on belül egy új <jms-queue> -tag-et az alábbiak szerint:

```
<subsystem xmlns="urn:jboss:domain:messaging-activemq:13.1">
    <server name="default">
        <!-- a <jms-queue> tag-ek elé tegyük be az új üzenetsort:
        <jms-queue name="TaskQueue" entries="TaskQueue
java:jboss/exported/jms/queue/TaskQueue"/>
```

A wildfly /bin/add-user.bat futtatásával létrehozhatunk egy felhasználót quser/Password_1 belépéssel a **guest** csoportban:

```
What type of user do you wish to add?
  a) Management User (mgmt-users.properties)
  b) Application User (application-users.properties)
(a): b
Enter the details of the new user to add.
Using realm 'ApplicationRealm' as discovered from the existing property
files.
Username : quser
Password recommendations are listed below. To modify these restrictions
edit the add-user.properties configuration file.
- The password should be different from the username
- The password should not be one of the following restricted values
{root, admin, administrator}
- The password should contain at least 8 characters, 1 alphabetic
character(s), 1 digit(s), 1 non-alphanumeric symbol(s)
Password :
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma
separated list, or leave blank for none)[ ]: guest
```

```
About to add user 'quser' for realm 'ApplicationRealm'
Is this correct yes/no? yes
Added user 'quser' to file
'C:\projects\devstudio\runtimes\wildfly-10.0.0.Final\standalone\configuration\application-users.properties'
Added user 'quser' to file
'C:\projects\devstudio\runtimes\wildfly-10.0.0.Final\domain\configuration\application-users.properties'
Added user 'quser' with groups guest to file
'C:\projects\devstudio\runtimes\wildfly-10.0.0.Final\standalone\configuration\application-roles.properties'
Added user 'quser' with groups guest to file
'C:\projects\devstudio\runtimes\wildfly-10.0.0.Final\domain\configuration\application-roles.properties'
Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a
Remoting connection for server to server EJB calls.
yes/no? yes
To represent the user add the following to the server-identities
definition <secret value="UGFzc3dvcmRfMQ==" />
Press any key to continue . . .
```

Az MDB klienst 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 kliens kódja legyen a következő:

```
package org.ait;

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.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);
}
}
```

Indítás után a szervert konzol-ban megjelenik a "Hello world" szöveg.

Az üzenetsorok finombeállításairól itt olvashatunk részletesen:

<https://docs.jboss.org/author/display/WFLY/Messaging%20configuration.html>

Feladat

Készítsünk egy servletet Send néven, amely a <http://localhost:8080/Queue/Send> -el elküld egy TextMessage üzenetet a java:/jms/queue/TestQueue sorra. Továbbá készítsünk egy másik servletet Receive néven, amely <http://localhost:8080/Queue/Receive> hívással 5 másodpercig hallgat a fenti üzenet sorra és ha van üzenet akkor kiírja azt, ha nincs, akkor kiírja a "nincs új üzenete" szöveget.

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:uezenetsorok

Last update: **2023/05/04 16:01**

