**Howto quickly setup an interface among systems without coding and using an opensource software**

*Matteo Redaelli*

[*http://www.redaelli.org/matteo/*](http://www.redaelli.org/matteo/)

*matteo DOT redaelli AT gmail DOT com*

*July 2012*

*Version 1.3*



[Abstract](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.uy4nycflo5hg)

[Installation](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.r693tweho3cz)

[Setup](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.dpa41cc3ctf6)

[Proxy (optional)](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.8be03fai9jq4)

[Webconsole (optional)](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.2ynd9wyqb2eo)

[Apache ActiveMQ (not needed)](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.vmakupq1tpni)

[Apache Camel](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.ks622wb5805v)

[JDBC drivers](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.b8f42zcepvaj)

[My sample integration project](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.3o0dvyoov41z)

[Configuring Datasources](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.kcb66jeif3bl)

[Mysql Database](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.698tavcid3ma)

[SQL Server database](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.c8qa8l2m9xwn)

[Routes](https://docs.google.com/document/d/1mDz-Esy9TH86BmgAfO8fstKm7UoazXNtJMJl6gmWeKk/edit#heading=h.f7yn7esdaikz)****

**Abstract**

In this article I’ll show how to create in 10 minutes an integration between two databases:

* looking for uses in the database MOODLE (mysql) with missing attributes
* for each of that users retrieving the missing attributes from the database UPMS (m$ sql server) and then
* adding the missing attributes to the database MOODLE

I’ll use

* [Apache Karaf](http://karaf.apache.org/) 2.2.8
* [Apache Camel](http://camel.apache.org/) 2.10.0

under Linux (tested with Debian and Suse SLES)

**Installation**

Be sure to have Java 1.6 JDK installed (check it running java -version)

As suggested by <http://karaf.apache.org/manual/latest-2.2.x/quick-start.html> download, unzip and run bin/karaf

**Setup**

**Proxy (optional)**

If you are behind a proxy, set

  "org.ops4j.pax.url.mvn.proxySupport=true"

in the file

   etc/org.ops4j.pax.url.mvn.cfg

and create/update your $HOME/.m2/settings.xml with

<settings>

 <proxies>

  <proxy>

     <active>true</active>

     <protocol>http</protocol>

     <host>myproxy.redaelli.org</host>

     <port>80</port>

     <nonProxyHosts>localhost|\*.redaelli.org</nonProxyHosts>

   </proxy>

 </proxies>

</settings>

**Webconsole (optional)**

features:install webconsole

You can see the web console  opening in your browser the url

<http://localhost:8181/system/console/>

**Apache ActiveMQ (not needed)**

As suggested by <http://activemq.apache.org/osgi-integration.html>

features:chooseurl activemq

features:install activemq-blueprint

features:install activemq-web-console

activemq:create-broker --type blueprint   (needed or is a broker created by default?)

You can see the web console opening in your browser the url

<http://localhost:8181/activemqweb/>

**Apache Camel**

As suggested by <http://camel.apache.org/karaf.html>

features:chooseurl camel 2.10.0

features:install camel

features:install camel-blueprint

features:install camel-jdbc

features:install camel-quartz

**JDBC drivers**

osgi:install -s ’wrap:mvn:net.sourceforge.jtds/jtds/1.2.4$Import-Package=\*;resolution:=optional&Export-Package=\*;version=1.2.4;-noimport:=true’

osgi:install -s wrap:mvn:mysql/mysql-connector-java/5.1.18

**My sample integration project**

**Configuring Datasources**

**M$ SQL server database**

Create a file pir-upms-ds.xml and copy it to the deploy/ folder

<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">

    <bean class="net.sourceforge.jtds.jdbcx.JtdsDataSource" id="upms-ds">

            <property name="serverName" value="mysqlserver.redaelli.org"></property>

            <property name="databaseName" value="UPMS"></property>

            <property name="portNumber" value="1198"></property>

            <property name="user" value="XXX"></property>

            <property name="password" value="TTT"></property>

    </bean>

   <service ref="upms-ds" auto-export="interfaces">

     <service-properties>

    <entry key="osgi.jndi.service.name" value="upms-ds"></entry>

     </service-properties>

   </service>

</blueprint>

**MySQL database**

Create a file pir-llab-ds.xml and copy it to the deploy/ folder

<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">

       <bean class="com.mysql.jdbc.jdbc2.optional.MysqlDataSource" id="llab-ds">

               <property name="serverName" value="moodledb.redaelli.org"></property>

               <property name="databaseName" value="moodle"></property>

               <property name="port" value="3306"></property>

               <property name="user" value="moodle"></property>

               <property name="password" value="pwd"></property>

       </bean>

       <service ref="llab-ds" auto-export="interfaces">

         <service-properties>

           <entry key="osgi.jndi.service.name" value="llab-ds"></entry>

         </service-properties>

       </service>

</blueprint>

**Routes**

Create a file pir-llab-users-attributes.xml and copy it to the deploy folder

<?xml version="1.0" encoding="UTF-8"?>

<blueprint

   xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"

   xmlns:cm="http://aries.apache.org/blueprint/xmlns/blueprint-cm/v1.0.0"

   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

   xsi:schemaLocation="http://www.osgi.org/xmlns/blueprint/v1.0.0 http://www.osgi.org/xmlns/blueprint/v1.0.0/blueprint.xsd">

 <reference id="llab-ds" interface="javax.sql.DataSource" filter="(osgi.jndi.service.name=llab-ds)"/>

 <reference id="upms-ds" interface="javax.sql.DataSource" filter="(osgi.jndi.service.name=upms-ds)"/>

 <camelContext xmlns="http://camel.apache.org/schema/blueprint">

   <route>

   <!--

        SELECT USERS WITHOUT CID

   -->

     <from uri="direct:pir.llab.users.attributes.update" />

     <setBody>

       <simple>

          select

            username

          from

            mdl\_user

          where

            auth='ldap' and

            deleted=0 and

            idnumber=''

       </simple>

     </setBody>

     <to uri="log:pir.llab.users.attributes.update?level=INFO" />

     <to uri="jdbc:llab-ds" />

     <split>

       <simple>body</simple>

       <!--

            GET CID and other attributes for each USER

       -->

       <transform>

         <simple>

            select

              account,

              Qualification as qualification,

              CID as cid

            from CorporateUsers

            where

              UPPER(account)=UPPER('${in.body[username]}')

         </simple>

       </transform>

       <to uri="log:pir.llab.users.attributes.update?level=INFO" />

       <to uri="jdbc:upms-ds" />

       <split>

         <simple>body</simple>

         <transform>

           <simple>

             update mdl\_user set

               idnumber='${in.body[cid]}',

               aim='${in.body[qualification]}'

             where

               upper(username)=upper('${in.body[account]}')

           </simple>

         </transform>

         <to uri="log:pir.llab.users.attributes.update?level=INFO" />

         <to uri="jdbc:llab-ds" />

       </split>

     </split>

   </route>

   <!--

        SCHEDULED JOBS

   -->

   <route>

     <from uri="quartz://pir-llab/users-cid/?cron=0+0/5+12-18+?+\*+MON-FRI" />

     <to uri="direct:pir.llab.users.attributes.update" />

   </route>

 </camelContext>

</blueprint>

Now the job has been scheduled, look at the log file data/log/karaf.log

Good LUCK

Matteo