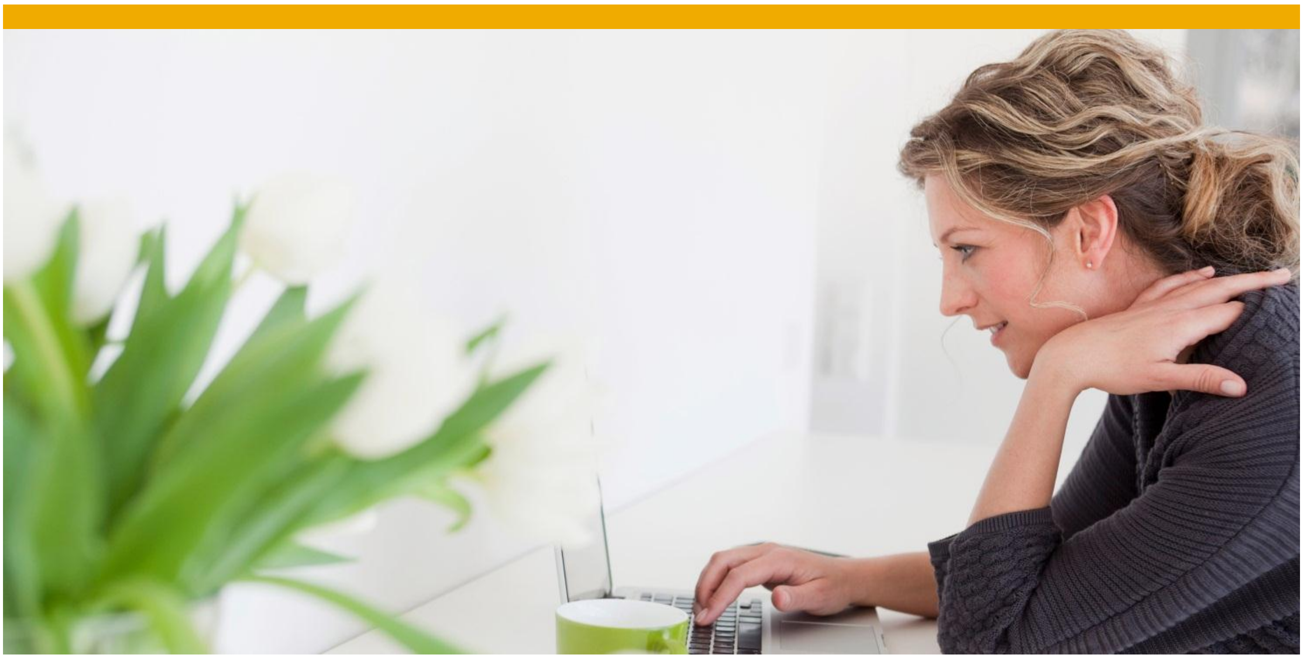


# LAB3a: Payment Handler Processing

Implement a New Payment Handler

# TABLE OF CONTENTS

**AGENDA** ..... 4
**BEFORE YOU START…** ..... 4
**IMPLEMENT THE PAYMENT HANDLER** ..... 4
**CONFIGURE THE PAYMENT HANDLER** ..... 7
**REPLACE THE STANDARD OFFLINE BANK ACCOUNT PAYMENT HANDLER** ..... 8
**BUILD PAYMENT HANDLER BUNDLE** ..... 8
**INSTALL PAYMENT HANDLER BUNDLE – HOT DEPLOYMENT** ..... 9
**TEST THE PAYMENT HANDLER** ..... 12

## AGENDA

1. Implement the Payment Handler
2. Configure the Payment Handler
3. Modify Distribution
4. Test the Payment Handler

## BEFORE YOU START…

Please start the Mobiliser 5.1 Lab Virtual machine.

The Login with user is "mobiliser" and password "sybase".

## IMPLEMENT THE PAYMENT HANDLER

The new payment handler will replace the default offline Bank Account handler. It simply logs the method calls issued by the payment broker. It will support both sides of a payment (debit and credit). To understand the concept behind the PaymentHandler architecture and implementation please refer to **“Money_Mobiliser_5.1_Customization_Guide.pdf” Section 3.3 Payment Handler.**

1. Navigate to com.sybase365.mobiliser.custom.project.businesslogic.impl bundle project in Eclipse.
2. Create a new Class (e.g. CustomBankAccountPaymentHandler.java) in following package:
`com.sybase365.mobiliser.custom.project.handlers.payment`
3. Implement the following interfaces:
com.sybase365.mobiliser.money.businesslogic.payment.api.IPaymentInstrumentClassHandler,
org.springframework.beans.factory.InitializingBean

Let Eclipse create the implemented methods – the class skeleton should look similar to what follows:

```
package com.sybase365.mobiliser.custom.project.handlers.payment;

import org.springframework.beans.factory.InitializingBean;
import
com.sybase365.mobiliser.money.businesslogic.payment.api.IPaymentInstrumentClassHandler;
import
com.sybase365.mobiliser.money.businesslogic.payment.api.exception.PaymentHandlerException;
import com.sybase365.mobiliser.money.businesslogic.payment.api.types.Balance;
import
com.sybase365.mobiliser.money.businesslogic.payment.api.types.PaymentInstrumentClassCombinati
on;
import com.sybase365.mobiliser.money.businesslogic.payment.api.types.PaymentMode;
import com.sybase365.mobiliser.money.businesslogic.payment.api.types.TransactionData;
import com.sybase365.mobiliser.money.persistence.model.pi.PaymentInstrument;

public class CustomBankAccountPaymentHandler implements
         IPaymentInstrumentClassHandler, InitializingBean {

  @Override
  public void authorise(TransactionData txnData, PaymentMode paymentMode)
              throws PaymentHandlerException {
       // TODO: Auto-generated method stub
  }

  @Override
  public Balance balanceInquiry(PaymentInstrument paymentInstrument)
              throws UnsupportedOperationException, PaymentHandlerException {
       // TODO: Auto-generated method stub
  }

(...)

  @Override
  public void afterPropertiesSet() throws Exception {
       // TODO: Auto-generated method stub
  }

  @Override
  public PaymentInstrumentClassCombination[] getCoverage() () {
       // TODO: Auto-generated method stub
  }
```

4. Modify the coverage to register for [bank, null] and [null, bank]

```
(...)

@Override
public PaymentInstrumentClassCombination[] getCoverage() {
   return new PaymentInstrumentClassCombination[] { new
PaymentInstrumentClassCombination(PaymentInstrument.Class.BANK, null),
   new PaymentInstrumentClassCombination(null, PaymentInstrument.Class.BANK)};
}
(...)
```

5. Add piLookup as member variable which would get injected using Spring. Implement afterPropertiesSet() method to throw IllegalStateException is Spring injection is unsuccessful.
6. Also don’t forget to add LOG static final variable – will be used to log information to the log files. Use slf4j packages for this.
7. Import the following package:

com.sybase365.mobiliser.money.businesslogic.payment.api.IPaymentInstrumentLookup

```
(...)

import com.sybase365.mobiliser.money.businesslogic.payment.api.IPaymentInstrumentLookup;

public class CustomBankAccountPaymentHandler implements
         IPaymentInstrumentClassHandler, InitializingBean {

      /** The <code>Log</code> instance to use. */
      private static final Logger LOG = LoggerFactory
           .getLogger(CustomBankAccountPaymentHandler.class);

       private IPaymentInstrumentLookup piLookup;

(...)
```

```
(...)

@Override
public void afterPropertiesSet() throws Exception {
   if (this.piLookup == null) {
      throw new IllegalStateException("piLookup must be set");
   }
}

(...)
```

8. Add getter and setters for the member variables so that Spring could inject the object instances
9. Add return value to getName()

```
(...)

    public IPaymentInstrumentLookup getPiLookup() {
        return this.piLookup;
    }

    public void setPiLookup(IPaymentInstrumentLookup piLookup) {
        this.piLookup = piLookup;
    }

(...)

    public String getName() {
        return "CustomBankAccountPaymentHandler";
    }
```

10. Add log statements to all method stubs (that take `TransactionData` and `PaymentMode` as input parameters). Log the account number (using the payment instrument lookup service) and whether it is a credit or debit call.
11. Import the following packages:

com.sybase365.mobiliser.money.persistence.model.pi.BankAccount
com.sybase365.mobiliser.money.persistence.model.transaction.Transaction

```
(...)

       Transaction mainTxn = txnData.getTransaction().getTransaction();
       BankAccount ba = null;
       if (paymentMode.equals(PaymentMode.CREDIT)) {
              ba = (BankAccount) this.piLookup.getPaymentInstrumentDelegateType(
                            mainTxn.getPaymentInstrumentPayee());
       } else {
               ba = (BankAccount) this.piLookup.getPaymentInstrumentDelegateType(
                            mainTxn.getPaymentInstrumentPayer());
       }
       LOG.info(" ############################################## "
              + "Successfully called <METHOD-NAME> using Bank Account #{}!"
              + " ##############################################",
       ba.getAccountNumber());

(...)
```

12. Change the text ‘METHOD-NAME’ to be the name of the method where this is inserted.
13. Format the code: “Ctrl+A” then “Ctrl+Shift+F” to format the contents of the file.
14. Organize the imports: “Ctrl+Shift+O” then verify the imports.
15. Save the CustomBankAccountPaymentHandler.java

## CONFIGURE THE PAYMENT HANDLER

Now define a new payment handler as a Spring bean and then export it to the OSGi service registry for transactions to make use of it during payment processing.

### bundle-context.xml

```
(...)

<bean id="customBankAccountPaymentHandler"
class="com.sybase365.mobiliser.custom.project.handlers.payment.CustomBankAccountPaymentHandle
r" >
     <property name="piLookup" ref="piLookup" />
</bean>

(...)
```

1. Please indent the XML contents by pressing “Ctrl+A” and then “Ctrl+I”.
2. Save the file **bundle-context.xml.**

### bundle-context-osgi.xml

```
(...)

<osgi:reference id="piLookup"
interface="com.sybase365.mobiliser.money.businesslogic.payment.api.IPaymentInstrumentLookup"
/>

<osgi:service
interface="com.sybase365.mobiliser.money.businesslogic.payment.api.IPaymentInstrumentClassHan
dler" ref="customBankAccountPaymentHandler" >
</osgi:service>

(...)
```

3. Please indent the XML contents by pressing “Ctrl+A” and then “Ctrl+I”.
4. Save the file **bundle-context-osgi.xml.**

## REPLACE THE STANDARD OFFLINE BANK ACCOUNT PAYMENT HANDLER

Open the **dist.xml** in the distribution module and go to the exclude section. Add the bank offline bank account payment handler back to the exclude list.

```
(...)

<!-- money -->
<fileSet>
   <directory>target/money/com.sybase365.mobiliser.vanilla.${container.type}-
${version.money}</directory>
   <outputDirectory>/money</outputDirectory>
   <fileMode>0644</fileMode>
   <directoryMode>0755</directoryMode>
   <excludes>
      <exclude>**/bin/*</exclude>
      <!-- exclude bundles which are replaced by custom implementations -->
      <exclude>**/bundles/*/*handlers.offline*</exclude>
      <exclude>**/bundles/*/*money.ams.export*</exclude>
      <exclude>**/bundles/*/*dummyreceiver*</exclude>
      <!--
       uncomment this if you need embedded system properties in preferences you'll
       also need to uncomment the patched bundle below and include it in the pom as
       a dependency. (as well as actually building the patched jar in the reactor)
       <exclude>**/bundles/*/*prefs.util*</exclude>
      -->
   </excludes>
</fileSet>

(...)
```

1. Please indent the XML contents by pressing “Ctrl+A” and then “Ctrl+I”.
2. Save the file **dist.xml.**

## BUILD PAYMENT HANDLER BUNDLE

1. Since we only modified businesslogic.impl bundle therefore, we should learn how to build an individual bundle and update it without bringing the Mobiliser money server down. (NOTE: The key is to know the bundles that have been updated – so that only those bundles get built and deployed to runtime environment – this hot deployment feature could greatly help development as well as release phases).
2. Open a terminal (Applications/Accessories/Terminal)
3. Change to the following directory
   ```
   $ cd ~/workspace/custom/businesslogic/impl
   ```
4. Run following command to build the new businesslogic/impl bundle – this bundle includes the payment handler.
   ```
   $ mvn clean install
   ```
5. Once the maven shows “**BUILD SUCCESS**”, go to the following directory:
   ```
   $ cd ~/workspace/custom/businesslogic/impl/target
   ```
6. You should see following two files among others:
   ```
   $ ls -l
   com.sybase365.mobiliser.custom.project.businesslogic.impl-1.2.0-SNAPSHOT.jar
   com.sybase365.mobiliser.custom.project.businesslogic.impl-1.2.0-SNAPSHOT-sources.jar
   ```

*com.sybase365.mobiliser.custom.project.businesslogic.impl-1.2.0-SNAPSHOT.jar* is the bundle file which we are interested in deploying.

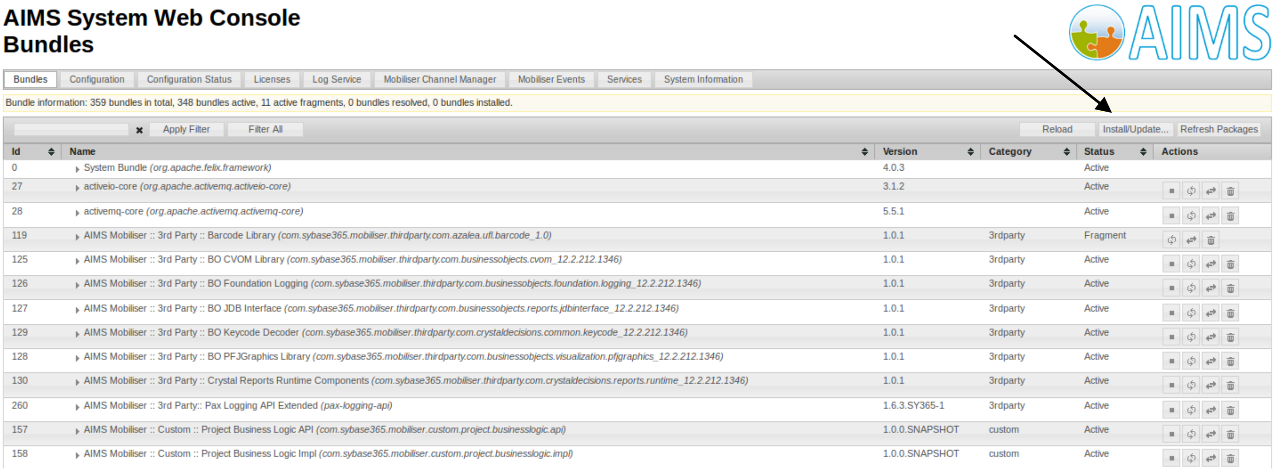
## INSTALL PAYMENT HANDLER BUNDLE – HOT DEPLOYMENT

Bundle Build directory:
~/workspace/custom/businesslogic/impl/target
Runtime Bundle Source directory:
~/workspace/custom/dist/target/com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT/money/bundles

There are two ways of updating the bundle:

- First: Apache Felix Web Console

  Apache Felix Web console has an option “Install/Update…” a bundle while the system is running. Even though this option is great but this option only installs to memory (server runtime environment), in other words if the system gets restarted the change made using “Install/Update…” option would get lost and the server would load the previous copy of the bundle.



**AIMS System Web Console**
**Bundles**

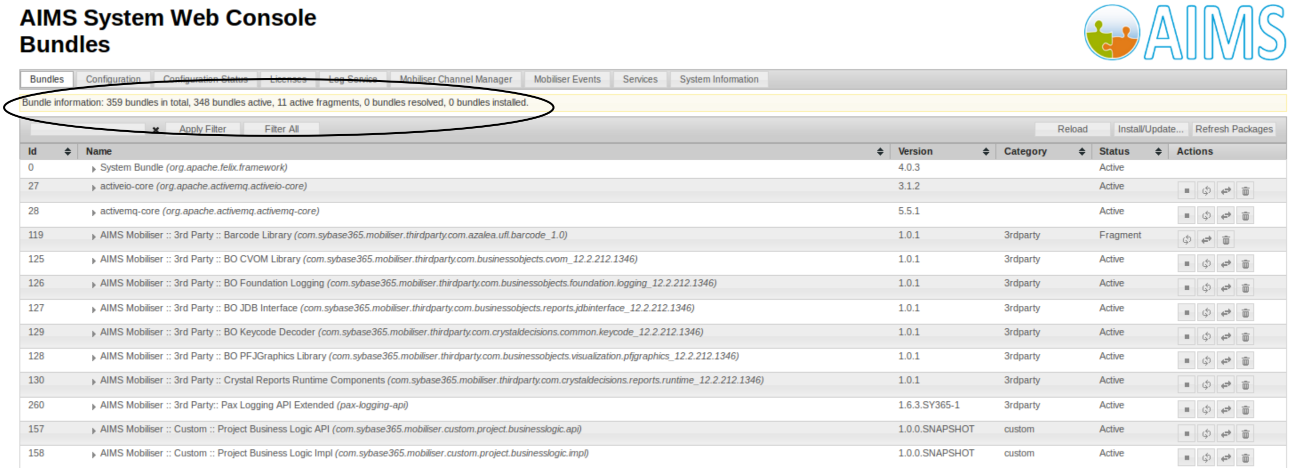
Bundle information: 359 bundles in total, 348 bundles active, 11 active fragments, 0 bundles resolved, 0 bundles installed.

| Id | Name | Version | Category | Status | Actions |
|---|---|---|---|---|---|
| 0 | System Bundle *(org.apache.felix.framework)* | 4.0.3 | | Active | |
| 27 | activeio-core *(org.apache.activemq.activeio-core)* | 3.1.2 | | Active | ■ ↻ ⇄ 🗑 |
| 28 | activemq-core *(org.apache.activemq.activemq-core)* | 5.5.1 | | Active | ■ ↻ ⇄ 🗑 |
| 119 | AIMS Mobiliser :: 3rd Party :: Barcode Library *(com.sybase365.mobiliser.thirdparty.com.azalea.ufl.barcode_1.0)* | 1.0.1 | 3rdparty | Fragment | ↻ ⇄ 🗑 |
| 125 | AIMS Mobiliser :: 3rd Party :: BO CVOM Library *(com.sybase365.mobiliser.thirdparty.com.businessobjects.cvom_12.2.212.1346)* | 1.0.1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 126 | AIMS Mobiliser :: 3rd Party :: BO Foundation Logging *(com.sybase365.mobiliser.thirdparty.com.businessobjects.foundation.logging_12.2.212.1346)* | 1.0.1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 127 | AIMS Mobiliser :: 3rd Party :: BO JDB Interface *(com.sybase365.mobiliser.thirdparty.com.businessobjects.reports.jdbinterface_12.2.212.1346)* | 1.0.1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 129 | AIMS Mobiliser :: 3rd Party :: BO Keycode Decoder *(com.sybase365.mobiliser.thirdparty.com.crystaldecisions.common.keycode_12.2.212.1346)* | 1.0.1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 128 | AIMS Mobiliser :: 3rd Party :: BO PFJGraphics Library *(com.sybase365.mobiliser.thirdparty.com.businessobjects.visualization.pfjgraphics_12.2.212.1346)* | 1.0.1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 130 | AIMS Mobiliser :: 3rd Party :: Crystal Reports Runtime Components *(com.sybase365.mobiliser.thirdparty.com.crystaldecisions.reports.runtime_12.2.212.1346)* | 1.0.1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 260 | AIMS Mobiliser :: 3rd Party:: Pax Logging API Extended *(pax-logging-api)* | 1.6.3.SY365-1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 157 | AIMS Mobiliser :: Custom :: Project Business Logic API *(com.sybase365.mobiliser.custom.project.businesslogic.api)* | 1.0.0.SNAPSHOT | custom | Active | ■ ↻ ⇄ 🗑 |
| 158 | AIMS Mobiliser :: Custom :: Project Business Logic Impl *(com.sybase365.mobiliser.custom.project.businesslogic.impl)* | 1.0.0.SNAPSHOT | custom | Active | ■ ↻ ⇄ 🗑 |

- Second: Copy the bundle into Runtime source directory (Recommended)
  Once the bundle is built we can copy the bundle into Runtime Bundle Source directory – this is the directory where Apache Felix would pick up the bundles from when the bundle gets updated or the server gets restarted. Following are the steps to follow:

  1. Businesslogic Impl bundle is available here ~/workspace/custom/businesslogic/impl/target
  2. $cd ~/workspace/custom
  3. Copy businesslogic.impl bundle into Runtime Bundle Source directory, directory for businesslogic.impl bundle is ***15-mobiliser-money-businesslogic-services***.
  4. Copying/pasting following command may cause additional CR/LF.
  5. $cp businesslogic/impl/target/com.sybase365.mobiliser.custom.project.businesslogic.impl-1.2.0-SNAPSHOT.jar  dist/target/com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT/money/bundles/15-mobiliser-money-businesslogic-services
  6. Log on to AIMS console http://localhost:8080/system/console/bundles
  7. Locate "businesslogic.impl" bundle – Stop the bundle – Update the bundle – Start the bundle ☺ Voila – please see snapshot of the AIMS web console for reference.

Launching AIMS web console (Apache Felix Console)
See 0 bundles resolved, 0 resolved (an indication that server started with no errors)

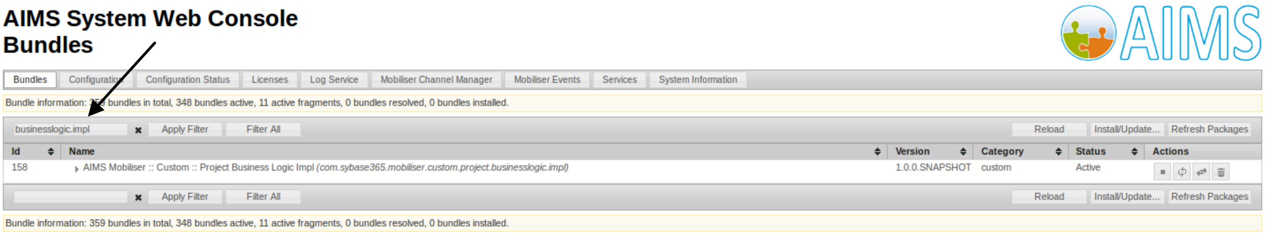


**AIMS System Web Console**
**Bundles**

Bundle information: 359 bundles in total, 348 bundles active, 11 active fragments, 0 bundles resolved, 0 bundles installed.

| Id | Name | Version | Category | Status | Actions |
|---|---|---|---|---|---|
| 0 | System Bundle *(org.apache.felix.framework)* | 4.0.3 | | Active | |
| 27 | activeio-core *(org.apache.activemq.activeio-core)* | 3.1.2 | | Active | ■ ↻ ⇄ 🗑 |
| 28 | activemq-core *(org.apache.activemq.activemq-core)* | 5.5.1 | | Active | ■ ↻ ⇄ 🗑 |
| 119 | AIMS Mobiliser :: 3rd Party :: Barcode Library *(com.sybase365.mobiliser.thirdparty.com.azalea.ufl.barcode_1.0)* | 1.0.1 | 3rdparty | Fragment | ↻ ⇄ 🗑 |
| 125 | AIMS Mobiliser :: 3rd Party :: BO CVOM Library *(com.sybase365.mobiliser.thirdparty.com.businessobjects.cvom_12.2.212.1346)* | 1.0.1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 126 | AIMS Mobiliser :: 3rd Party :: BO Foundation Logging *(com.sybase365.mobiliser.thirdparty.com.businessobjects.foundation.logging_12.2.212.1346)* | 1.0.1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 127 | AIMS Mobiliser :: 3rd Party :: BO JDB Interface *(com.sybase365.mobiliser.thirdparty.com.businessobjects.reports.jdbinterface_12.2.212.1346)* | 1.0.1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 129 | AIMS Mobiliser :: 3rd Party :: BO Keycode Decoder *(com.sybase365.mobiliser.thirdparty.com.crystaldecisions.common.keycode_12.2.212.1346)* | 1.0.1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 128 | AIMS Mobiliser :: 3rd Party :: BO PFJGraphics Library *(com.sybase365.mobiliser.thirdparty.com.businessobjects.visualization.pfjgraphics_12.2.212.1346)* | 1.0.1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 130 | AIMS Mobiliser :: 3rd Party :: Crystal Reports Runtime Components *(com.sybase365.mobiliser.thirdparty.com.crystaldecisions.reports.runtime_12.2.212.1346)* | 1.0.1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 260 | AIMS Mobiliser :: 3rd Party:: Pax Logging API Extended *(pax-logging-api)* | 1.6.3.SY365-1 | 3rdparty | Active | ■ ↻ ⇄ 🗑 |
| 157 | AIMS Mobiliser :: Custom :: Project Business Logic API *(com.sybase365.mobiliser.custom.project.businesslogic.api)* | 1.0.0.SNAPSHOT | custom | Active | ■ ↻ ⇄ 🗑 |
| 158 | AIMS Mobiliser :: Custom :: Project Business Logic Impl *(com.sybase365.mobiliser.custom.project.businesslogic.impl)* | 1.0.0.SNAPSHOT | custom | Active | ■ ↻ ⇄ 🗑 |

Locating "businesslogic.impl" bundle (use the filter text box – to filter the bundle)
Typing "businesslogic.impl" in the mentioned text box and pressing "Enter" would bring the bundle



**AIMS System Web Console**
**Bundles**

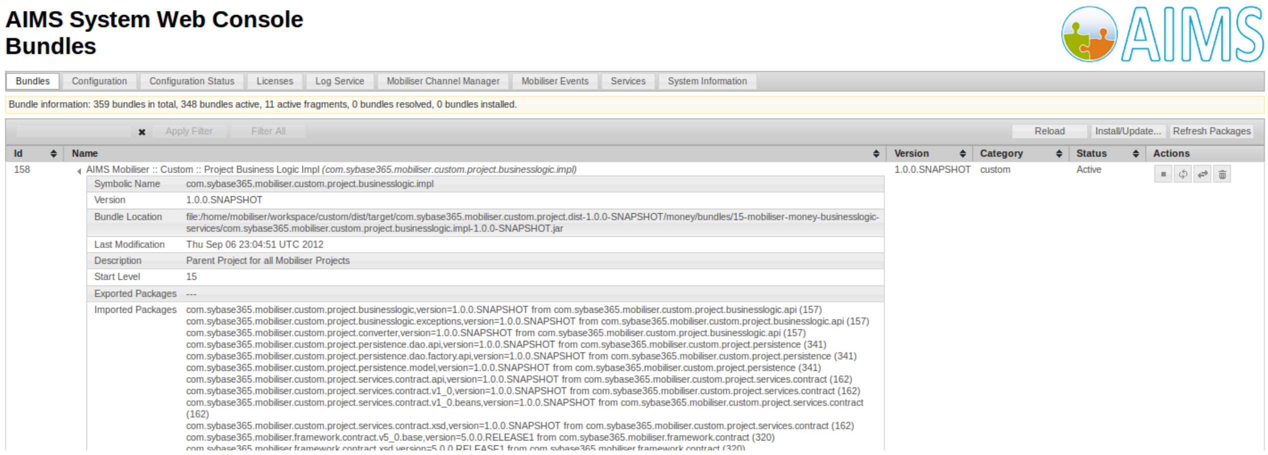
Bundle information: 359 bundles in total, 348 bundles active, 11 active fragments, 0 bundles resolved, 0 bundles installed.

| Id | Name | Version | Category | Status | Actions |
|---|---|---|---|---|---|
| 158 | AIMS Mobiliser :: Custom :: Project Business Logic Impl *(com.sybase365.mobiliser.custom.project.businesslogic.impl)* | 1.0.0.SNAPSHOT | custom | Active | ■ ↻ ⇄ 🗑 |

Bundle information: 359 bundles in total, 348 bundles active, 11 active fragments, 0 bundles resolved, 0 bundles installed.

## Navigate to the bundle and select it from the list

Click on contents under “Name” column would bring up bundle details page.



AIMS System Web Console
Bundles

Bundles | Configuration | Configuration Status | Licenses | Log Service | Mobiliser Channel Manager | Mobiliser Events | Services | System Information

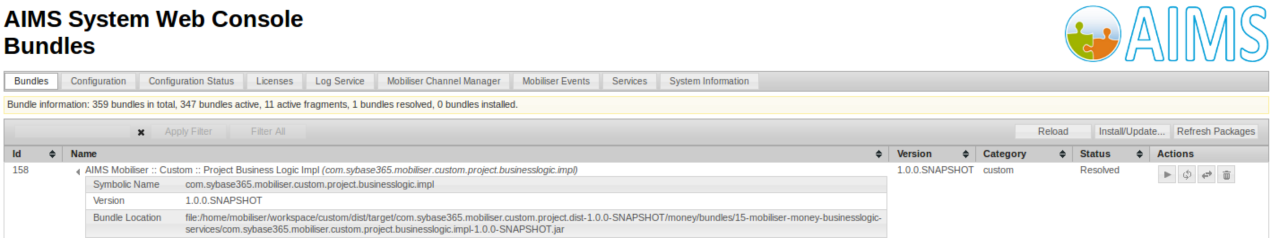
Bundle information: 359 bundles in total, 348 bundles active, 11 active fragments, 0 bundles resolved, 0 bundles installed.

| Id | Name | Version | Category | Status | Actions |
|---|---|---|---|---|---|
| 158 | AIMS Mobiliser :: Custom :: Project Business Logic Impl *(com.sybase365.mobiliser.custom.project.businesslogic.impl)* | 1.0.0.SNAPSHOT | custom | Active | ■ |

| | |
|---|---|
| Symbolic Name | com.sybase365.mobiliser.custom.project.businesslogic.impl |
| Version | 1.0.0.SNAPSHOT |
| Bundle Location | file:/home/mobiliser/workspace/custom/dist/target/com.sybase365.mobiliser.custom.project.dist-1.0.0-SNAPSHOT/money/bundles/15-mobiliser-money-businesslogic-services/com.sybase365.mobiliser.custom.project.businesslogic.impl-1.0.0-SNAPSHOT.jar |
| Last Modification | Thu Sep 06 23:04:51 UTC 2012 |
| Description | Parent Project for all Mobiliser Projects |
| Start Level | 15 |
| Exported Packages | --- |
| Imported Packages | com.sybase365.mobiliser.custom.project.businesslogic,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.businesslogic.api (157)<br>com.sybase365.mobiliser.custom.project.businesslogic.exceptions,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.businesslogic.api (157)<br>com.sybase365.mobiliser.custom.project.converter,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.businesslogic.api (157)<br>com.sybase365.mobiliser.custom.project.persistence.dao.api,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.persistence (341)<br>com.sybase365.mobiliser.custom.project.persistence.dao.factory.api,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.persistence (341)<br>com.sybase365.mobiliser.custom.project.persistence.model,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.persistence (341)<br>com.sybase365.mobiliser.custom.project.services.contract.api,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.services.contract (162)<br>com.sybase365.mobiliser.custom.project.services.contract.v1_0,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.services.contract (162)<br>com.sybase365.mobiliser.custom.project.services.contract.v1_0.beans,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.services.contract (162)<br>com.sybase365.mobiliser.custom.project.services.contract.xsd,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.services.contract (162)<br>com.sybase365.mobiliser.framework.contract.v5_0.base,version=5.0.0.RELEASE1 from com.sybase365.mobiliser.framework.contract (320)<br>com.sybase365.mobiliser.framework.contract.xsd,version=5.0.0.RELEASE1 from com.sybase365.mobiliser.framework.contract (320) |

## Stop the Bundle

Press “Stop” button (under Actions), bundle status would get changed to “Resolved” (refresh the page).



AIMS System Web Console
Bundles

Bundles | Configuration | Configuration Status | Licenses | Log Service | Mobiliser Channel Manager | Mobiliser Events | Services | System Information

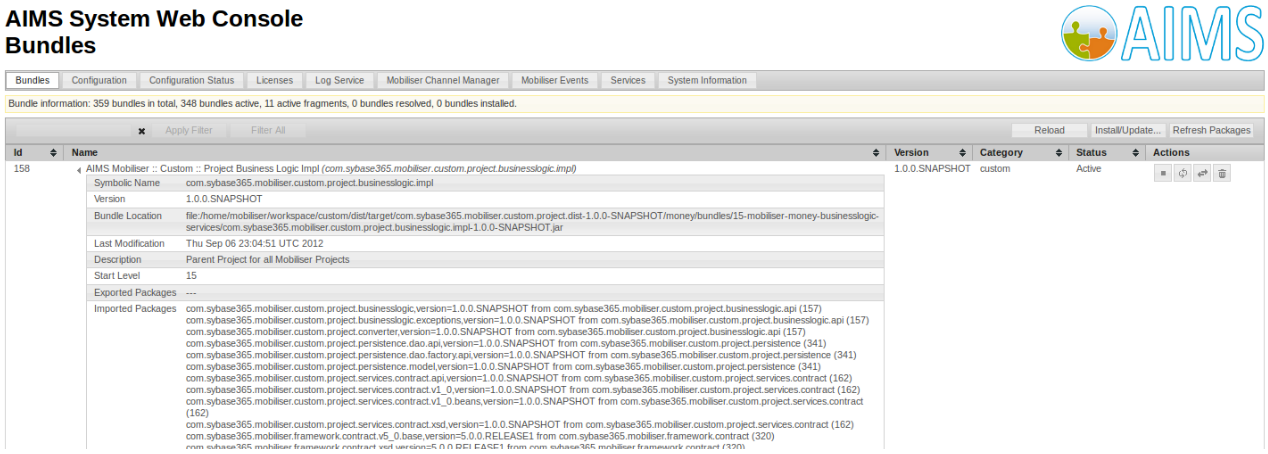
Bundle information: 359 bundles in total, 347 bundles active, 11 active fragments, 1 bundles resolved, 0 bundles installed.

| Id | Name | Version | Category | Status | Actions |
|---|---|---|---|---|---|
| 158 | AIMS Mobiliser :: Custom :: Project Business Logic Impl *(com.sybase365.mobiliser.custom.project.businesslogic.impl)* | 1.0.0.SNAPSHOT | custom | Resolved | ▶ |

| | |
|---|---|
| Symbolic Name | com.sybase365.mobiliser.custom.project.businesslogic.impl |
| Version | 1.0.0.SNAPSHOT |
| Bundle Location | file:/home/mobiliser/workspace/custom/dist/target/com.sybase365.mobiliser.custom.project.dist-1.0.0-SNAPSHOT/money/bundles/15-mobiliser-money-businesslogic-services/com.sybase365.mobiliser.custom.project.businesslogic.impl-1.0.0-SNAPSHOT.jar |

## Update the Bundle

Click on “Update” button (under Actions), will load the new contents of the bundle from the Runtime Source directory – if everything works out fine the bundle status will changed to “Installed” after you refresh the page. Once the bundle is in “Installed” state (all the package dependencies have been resolved) it can be started and becomes “Active” after successful startup (refresh the page).



AIMS System Web Console
Bundles

Bundles | Configuration | Configuration Status | Licenses | Log Service | Mobiliser Channel Manager | Mobiliser Events | Services | System Information

Bundle information: 359 bundles in total, 348 bundles active, 11 active fragments, 0 bundles resolved, 0 bundles installed.

| Id | Name | Version | Category | Status | Actions |
|---|---|---|---|---|---|
| 158 | AIMS Mobiliser :: Custom :: Project Business Logic Impl *(com.sybase365.mobiliser.custom.project.businesslogic.impl)* | 1.0.0.SNAPSHOT | custom | Active | ■ |

| | |
|---|---|
| Symbolic Name | com.sybase365.mobiliser.custom.project.businesslogic.impl |
| Version | 1.0.0.SNAPSHOT |
| Bundle Location | file:/home/mobiliser/workspace/custom/dist/target/com.sybase365.mobiliser.custom.project.dist-1.0.0-SNAPSHOT/money/bundles/15-mobiliser-money-businesslogic-services/com.sybase365.mobiliser.custom.project.businesslogic.impl-1.0.0-SNAPSHOT.jar |
| Last Modification | Thu Sep 06 23:04:51 UTC 2012 |
| Description | Parent Project for all Mobiliser Projects |
| Start Level | 15 |
| Exported Packages | --- |
| Imported Packages | com.sybase365.mobiliser.custom.project.businesslogic,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.businesslogic.api (157)<br>com.sybase365.mobiliser.custom.project.businesslogic.exceptions,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.businesslogic.api (157)<br>com.sybase365.mobiliser.custom.project.converter,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.businesslogic.api (157)<br>com.sybase365.mobiliser.custom.project.persistence.dao.api,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.persistence (341)<br>com.sybase365.mobiliser.custom.project.persistence.dao.factory.api,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.persistence (341)<br>com.sybase365.mobiliser.custom.project.persistence.model,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.persistence (341)<br>com.sybase365.mobiliser.custom.project.services.contract.api,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.services.contract (162)<br>com.sybase365.mobiliser.custom.project.services.contract.v1_0,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.services.contract (162)<br>com.sybase365.mobiliser.custom.project.services.contract.v1_0.beans,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.services.contract (162)<br>com.sybase365.mobiliser.custom.project.services.contract.xsd,version=1.0.0.SNAPSHOT from com.sybase365.mobiliser.custom.project.services.contract (162)<br>com.sybase365.mobiliser.framework.contract.v5_0.base,version=5.0.0.RELEASE1 from com.sybase365.mobiliser.framework.contract (320)<br>com.sybase365.mobiliser.framework.contract.xsd,version=5.0.0.RELEASE1 from com.sybase365.mobiliser.framework.contract (320) |

Once you refresh, make sure “customBankAccountPaymentHandler” appears as a service. This will make sure that payment handler service is started successfully. Be aware that it might be in IPaymentInstrumentClassHandler.

Alternatively JConsole can be used to locate the new Payment Handler as well.

## TEST THE PAYMENT HANDLER

In order for the logging output to occur there needs to be a change made to the logging configuration.

1. Navigate to ~/workspace/custom/dist/target/com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT/money/conf/cfgbackup/
2. Edit org.ops4j.pax.logging.properties by adding (right below `log4j.rootCategory=WARN, S`):

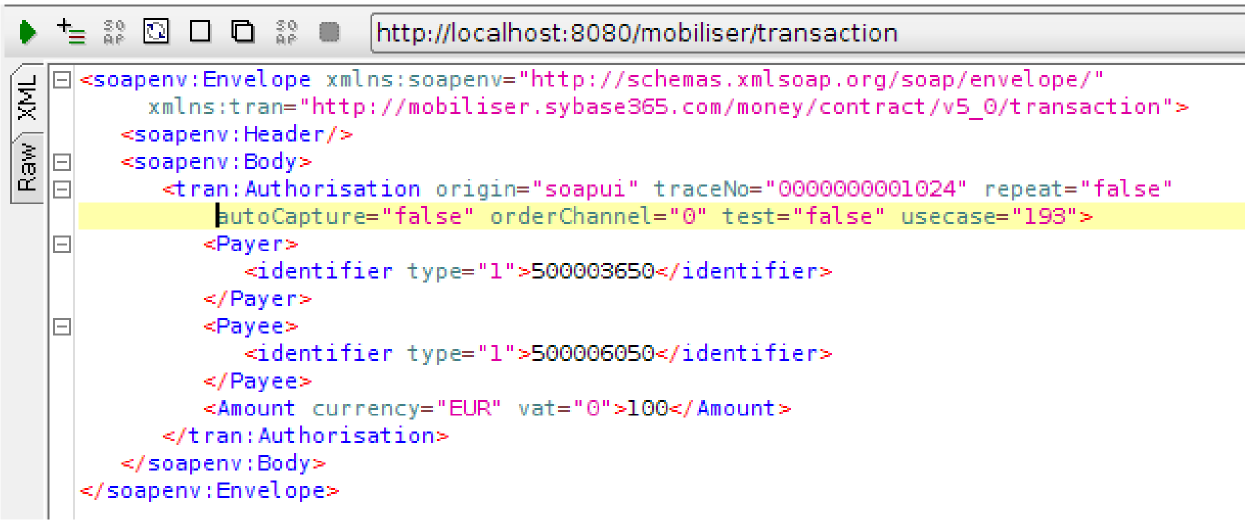
   ```
   log4j.logger.com.sybase365.mobiliser.custom=DEBUG
   ```

3. Save the file and move it to~/workspace/custom/dist/target/com.sybase365.mobiliser.custom.project.dist-1.2.0-SNAPSHOT/money/conf/cfgload.
4. The container will automatically load the properties from cfgload folder. No restarts are required.

Start the portal and (if not done before) and perform following tasks:

1. Make sure that the Web Portal comes up with no errors in the log file
2. Signup a Consumer (You should know by now – this exactly same way as was done as part of Lab1)
3. Add a Bank account (or select the existing one)
4. Transfer money from the Bank Account to the SVA.
5. Look for the log statement in …/money/logs/mobiliser.log.transaction

```
###############################################
Successfully called <METHOD-NAME> using Bank Account #<ID>!
###############################################
```



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tran="http://mobiliser.sybase365.com/money/contract/v5_0/transaction">
  <soapenv:Header/>
  <soapenv:Body>
    <tran:Authorisation origin="soapui" traceNo="0000000001024" repeat="false" autoCapture="false"
orderChannel="0" test="false" usecase="193">
      <Payer>
        <identifier type="1">500003650</identifier>
      </Payer>
      <Payee>
        <identifier type="1">500006050</identifier>
      </Payee>
      <Amount currency="EUR" vat="0">100</Amount>
    </tran:Authorisation>
  </soapenv:Body>
</soapenv:Envelope>
```

www.sap.com

© 2013 SAP AG. All rights reserved.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.