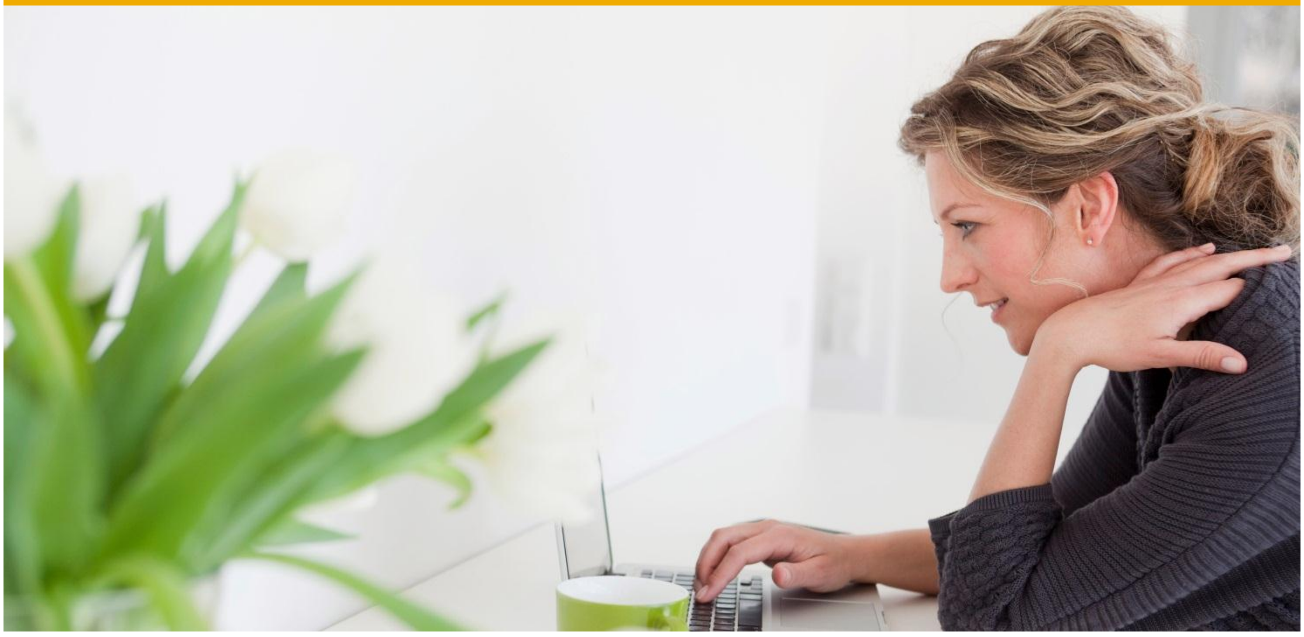


LAB4a: Event Handler Processing

Implement a New Event and Handler

TABLE OF CONTENTS

AGENDA

1. Implement the Event Class
2. Implement the Event Handler
3. Configure the Event Handler
4. Create an Event
5. Test the Event Handler

BEFORE YOU START…

Please start the Mobiliser 5.1 Lab Virtual machine.

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

IMPLEMENT THE EVENT CLASS

The "DepartmentEvent" event class accesses data pertaining to event handling. See Chapter 9 in the Mobiliser_Framework_5.1_Development_Guide.pdf.

1. Navigate to com.sybase365.mobiliser.custom.project.businesslogic.impl bundle project in Eclipse.
2. Create a new event class **DepartmentEvent.java** implementing the com.sybase365.mobiliser.framework.event.model.CriteriaConditionalEvent, let Eclipse create the init() method to implement.
3. DepartmentEvent.java should be created in the following package: com.sybase365.mobiliser.custom.project.jobs.event.model

```java
package com.sybase365.mobiliser.custom.project.jobs.event.model;

import com.sybase365.mobiliser.framework.event.model.CriteriaConditionalEvent;
import com.sybase365.mobiliser.framework.event.model.data.EventCriteria;
import com.sybase365.mobiliser.framework.event.model.data.EventData;

public class DepartmentEvent extends CriteriaConditionalEvent{
    private static final long serialVersionUID = 1L;
    public static final String EVENT_TYPE = "DepartmentEvent";

    @Override
    public void init() {
       super.setName(EVENT_TYPE);
       if (this.getData() == null) {
           this.setData(new EventData());
       }
    }
}
```

4. Create the getters and setters for the event data (department ID and name). If you use Eclipse to generate the getters/setters, remove 'Key' from the method names.

```
(...)

private static final String KEY_DEPARTMENT_ID = "departmentId";
private static final String KEY_DEPARTMENT_NAME = "departmentName";

(...)

public long getDepartmentId() {
   return Long.parseLong(this.getData().get(KEY_DEPARTMENT_ID));
}

public void setDepartmentId(final long departmentId) {
   this.getData().put(KEY_DEPARTMENT_ID, Long.toString(departmentId));
}

public String getDepartmentName() {
   return this.getData().get(KEY_DEPARTMENT_NAME);
}

public void setDepartmentName(final String departmentName) {
   this.getData().put(KEY_DEPARTMENT_NAME, departmentName);
}

(...)
```

5. Modify the init method to set the event name and apply the criteria (only for departments created with name “TEST_DEPARTMENT”).

```
(...)

@Override
public void init() {
   super.setName(EVENT_TYPE);
   if (this.getData() == null) {
      this.setData(new EventData());
   }
   EventCriteria criteria = new EventCriteria().where(KEY_DEPARTMENT_NAME)
      .eq("TEST_DEPARTMENT");
   setCriteria(criteria);
}

(...)
```

6. Format the code: “Ctrl+A” then “Ctrl+Shift+F” to format the contents of the file.
7. Organize the imports: “Ctrl+Shift+O” then verify the imports.
8. Save the DepartmentEvent.java

IMPLEMENT THE EVENT HANDLER

The new event handler performs asynchronous logic based on the creation of a new department record in the database (DepartmentCreatedEvent). See 9.2.3/9.6 in the Mobiliser_Framework_5.1_Development_Guide.pdf.

1. Navigate to com.sybase365.mobiliser.custom.project.businesslogic.impl bundle project in Eclipse.
2. Create a new class **DepartmentEventHandler.java** implementing the `com.sybase365.mobiliser.money.jobs.event.handler.util.MoneyEventHandler`, let Eclipse creates the methods to implement.
3. DepartmentEventHandler.java should be created in the following package:
4. `com.sybase365.mobiliser.custom.project.jobs.event.handler.department`

```
package com.sybase365.mobiliser.custom.project.jobs.event.handler.department;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.sybase365.mobiliser.custom.project.jobs.event.model.DepartmentEvent;
import com.sybase365.mobiliser.custom.project.persistence.dao.factory.api.DaoFactory;
import com.sybase365.mobiliser.custom.project.persistence.model.Department;
import com.sybase365.mobiliser.framework.event.model.Event;
import com.sybase365.mobiliser.money.jobs.event.handler.util.MoneyEventHandler;
import com.sybase365.mobiliser.money.jobs.event.handler.util.MoneyEventHandlerConfiguration;
import com.sybase365.mobiliser.util.tools.formatutils.FormatUtils;

public class DepartmentEventHandler extends
       MoneyEventHandler<MoneyEventHandlerConfiguration> {

    private static final Logger LOG = LoggerFactory
           .getLogger(DepartmentEventHandler.class);
    private DaoFactory daoFactory;

    @Override
    public void afterPropertiesSet() throws Exception {
       if (this.daoFactory == null) {
           throw new IllegalStateException("daoFactory is required");
       }
       super.afterPropertiesSet();
    }

    @Override
    public String getHandlerName() {
       return "DepartmentEventHandler";
    }
    @Override
    public String getEventName() {
       return DepartmentEvent.EVENT_TYPE;
    }

    @Override
    public boolean process(Event e) {
       // Create event logic here
   }

    public void setDaoFactory(DaoFactory daoFactory) {
       this.daoFactory = daoFactory;
    }
}
```

5. Implement the logic to create a Department from an Event object passed as a parameter.

```
(...)

@Override
public boolean process(Event e) {
   DepartmentEvent departmentEvent = new DepartmentEvent();
   departmentEvent.setData(e.getData());
   try {
      Department department = this.daoFactory.getDepartmentDao().getById(
         Long.valueOf(departmentEvent.getDepartmentId()));
      department.setName(FormatUtils.formatString(
         departmentEvent.getDepartmentName(), 1, 10, false));
      this.daoFactory.getDepartmentDao().save(department, new Long(0));
   } catch (Exception e1) {
      LOG.debug("Update of department with ID {} failed",
      Long.valueOf(departmentEvent.getDepartmentId()));
      return false;
   }
   LOG.debug("Update of department with ID {} successful",
   Long.valueOf(departmentEvent.getDepartmentId()));
   return true;
}

(...)
```

6. Format the code: “Ctrl+A” then “Ctrl+Shift+F” to format the contents of the file.
7. Organize the imports: “Ctrl+Shift+O” then verify the imports.
8. Save DepartmentEventHandler.java

9. Create a configuration class extending the *com.sybase365.mobiliser.money.jobs.event.handler.util.MoneyEventHandlerConfiguration* in the following package: `com.sybase365.mobiliser.custom.project.jobs.event.handler.department`

```
package com.sybase365.mobiliser.custom.project.jobs.event.handler.department;

import com.sybase365.mobiliser.money.jobs.event.handler.util.MoneyEventHandlerConfiguration;
public class DepartmentEventHandlerConfiguration extends
              MoneyEventHandlerConfiguration {

    // Custom configuration would be added here
}
```

10. Format the code: “Ctrl+A” then “Ctrl+Shift+F” to format the contents of the file.
11. Organize the imports: “Ctrl+Shift+O” then verify the imports.
12. Save DepartmentEventHandlerConfiguration.java

See section ***9.8 Configuration in the Mobiliser_Framework_5.1_Development_Guide.pdf*** for more information.

POM FILE MODOFICATION

Modify the project bundle pom.xml file by adding an entry for the “com.sybase365.mobiliser.custom.project.jobs.event.handler.department” package. Add it under the "Private-Package" tag to protect the implementation details and because it is not required outside the scope of this bundle.

```
(...)

<Private-Package>
(...)
       com.sybase365.mobiliser.custom.project.jobs.event.handler.blacklist,
       com.sybase365.mobiliser.custom.project.jobs.event.handler.department
</Private-Package>
(...)
```

1. Copying/Pasting above xml will drop the indentation – please press “Ctrl+A” to select the contents and then press “Ctrl+I” for indentation of XML contents.
2. Save the file **pom.xml.**

CREATE AN EVENT

A new event can now be created anywhere in the business logic.

1. Create the new event. One of the possible places could be when the department is getting created (i.e. DepartmentLogicImpl.java in businesslogic.impl bundle)

```
(...)

import com.sybase365.mobiliser.custom.project.jobs.event.model.DepartmentEvent;
import com.sybase365.mobiliser.framework.event.model.data.EventDelay;
import com.sybase365.mobiliser.framework.event.generator.EventGenerator;

(...)

public class DepartmentLogicImpl implements IDepartmentLogic, InitializingBean {
    private DaoFactory daoFactory;
    protected EventGenerator eventGenerator;

    @Override
    public void afterPropertiesSet() throws Exception {
       if (this.daoFactory == null) {
           throw new IllegalStateException("daoFactory is required");
       }
       if (this.eventGenerator == null) {
           throw new IllegalStateException("eventGenerator is required");
       }
    }

    public EventGenerator getEventGenerator () {
       return this.eventGenerator;
    }

    public void setEventGenerator(EventGenerator eventGenerator) {
       this.eventGenerator = eventGenerator;
    }

    @Override
    public long createDepartment(Department department, long callerId) {
      getDaoFactory().getDepartmentDao().save(department, Long.valueOf(callerId));

      DepartmentEvent event = new DepartmentEvent();
      event.setDepartmentId(department.getId().longValue());
      event.setDepartmentName(department.getName());
      EventDelay oneMinuteDelay = new EventDelay();
      oneMinuteDelay.delayFor(1L).minutes();
      event.setDelay(oneMinuteDelay);

      this.eventGenerator.create(event);

      return department.getId().longValue();
   }

(...)
```

2. Format the code: “Ctrl+A” then “Ctrl+Shift+F” to format the contents of the file.
3. Organize the imports: “Ctrl+Shift+O” then verify the imports.
4. Save DepartmentLogicImpl.java

CONFIGURE THE EVENT HANDLER

Now add the new event handler to the service export of the customized event handlers. See 9.8 in the Mobiliser Framework Development Guide. Please do not forget to add the class/bean to the spring configuration. Also do not forget to inject the event handler lookup service injection.

bundle-context.xml

```
(...)

<bean id="departmentLogic"
class="com.sybase365.mobiliser.custom.project.businesslogic.impl.DepartmentLogicImpl">
    <property name="daoFactory" ref="daoFactory" />
    <property name="eventGenerator" ref="eventGenerator" />
</bean>

(...)

<bean id="departmentConfiguration"
class="com.sybase365.mobiliser.custom.project.jobs.event.handler.department.DepartmentEventHa
ndlerConfiguration">
  <property name="preferences" ref="prefsNode" />
</bean>

(...)

<bean id="departmentEventHandler"
   class="org.springframework.aop.framework.ProxyFactoryBean" >
     <property name="target" >
         <bean
class="com.sybase365.mobiliser.custom.project.jobs.event.handler.department.DepartmentEventHa
ndler" >
             <!-- Spring Injection -->
             <property name="daoFactory" ref="daoFactory" />
             <property name="eventGenerator" ref="eventGenerator" />
             <property name="configuration" ref="departmentConfiguration" />
         </bean>
     </property>
     <property name="interceptorNames" >
         <list>
             <value>eventTxAdvice</value>
         </list>
     </property>
     <property name="interfaces" >
         <list>
             <value>com.sybase365.mobiliser.framework.event.model.Handler</value>
         </list>
     </property>
</bean>

(...)
```

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

bundle-context-osgi.xml

```
(...)

<osgi:service
   interface="com.sybase365.mobiliser.framework.event.model.Handler"
   ref="departmentEventHandler" />

(...)
```

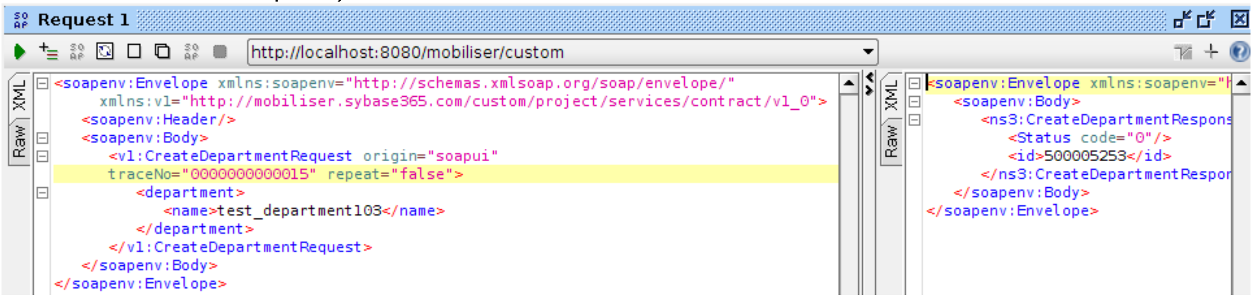
1. Please indent the XML contents by pressing “Ctrl+A” and then “Ctrl+I”.
2. Save the file **bundle-context-osgi.xml.**

BUILD THE BUNDLE

BusinessLogic.Impl is the only bundle modified for this lab. Build the bundle and use the "Hot Deployment" technique to deploy it. For instructions please refer to Lab 3a, "INSTALL PAYMENT HANDLER BUNDLE – HOT DEPLOYMENT" section

TEST THE EVENT HANDLER

1. Start SoapUI. If you saved the test from Lab 2B open it, else please create a new project.
2. Verify that you have access to WSDL.
3. In SoapUI change the request property values for “Authentication Type”=”Preemptive” and provide values for Username=mobiliser and Password=secret
4. We need to modify the contents of the ‘v1’ tag. Remove the following attributes: callback, conversationId, and sessionId
5. Give the origin tag the value of SoapUI and give traceNo a 6 digit number (Note each time you make a request you must change this value, it must be unique for each request).
6. Remove the following tags: Audit-Data and all sub-tags, UnstructuredData and all sub-tags, and department.id tag.
7. Create two departments, one named ‘TEST_DEPARTMENT’ the other can be any name you would like. Be aware that you must change the traceNo attribute for each request.
8. Look at money/logs/mobiliser.log.custom look for the log output to verify the test criteria was captured.
9. To create a department your SoapUI screen should look similar to the following: (Right pane shows the result of the request.)



http://localhost:8080/mobiliser/custom

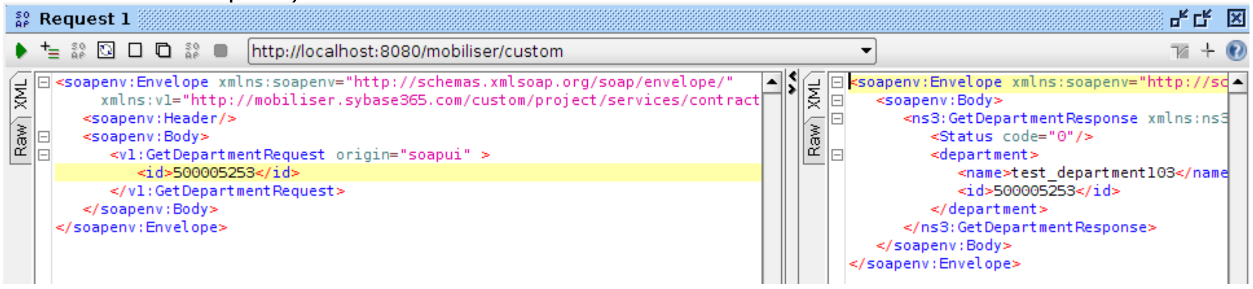
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:v1="http://mobiliser.sybase365.com/custom/project/services/contract/v1_0">
  <soapenv:Header/>
  <soapenv:Body>
    <v1:CreateDepartmentRequest origin="soapui"
    traceNo="0000000000015" repeat="false">
      <department>
        <name>test_department103</name>
      </department>
    </v1:CreateDepartmentRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope xmlns:soapenv="h
  <soapenv:Body>
    <ns3:CreateDepartmentRespons
      <Status code="0"/>
      <id>500005253</id>
    </ns3:CreateDepartmentRespor
  </soapenv:Body>
</soapenv:Envelope>
```

10. SoapUI code for creating a department named “test_department103”:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:v1="http://mobiliser.sybase365.com/custom/project/services/contract/v1_0">
  <soapenv:Header/>
  <soapenv:Body>
    <v1:CreateDepartmentRequest origin="soapui"
    traceNo="0000000000015" repeat="false">
      <department>
        <name>test_department103</name>
      </department>
    </v1:CreateDepartmentRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

11. To “get” a department your SoapUI screen should look similar to the following: (Right pane shows the result of the request.)



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:v1="http://mobiliser.sybase365.com/custom/project/services/contract
  <soapenv:Header/>
  <soapenv:Body>
    <v1:GetDepartmentRequest origin="soapui" >
      <id>500005253</id>
    </v1:GetDepartmentRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope xmlns:soapenv="http://sc
  <soapenv:Body>
    <ns3:GetDepartmentResponse xmlns:ns3
      <Status code="0"/>
      <department>
        <name>test_department103</name
        <id>500005253</id>
      </department>
    </ns3:GetDepartmentResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

12. SoapUI code to get a department name with a department id of “500005253”:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:v1="http://mobiliser.sybase365.com/custom/project/services/contract/v1_0">
  <soapenv:Header/>
  <soapenv:Body>
    <v1:GetDepartmentRequest origin="soapui" >
      <id>500005253</id>
    </v1:GetDepartmentRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

www.sap.com