
JOnES Project

Enterprise Bus: Demo v3 manual

AUTHORS: **G. VAUDAUX-RUTH (OW @ INRIA)**
CONTRIBUTORS: **F. RAVOAJANAHARY (OW @ INRIA)**
 S. QUAIREAU (OW @ INRIA)

Release:	19/03/2008
Status:	Release
Version:	1.0
Reference:	

1 INTRODUCTION.....	3
2 REQUIREMENTS.....	5
2.1 ESB.....	5
2.2 Mail Server (optional).....	5
2.3 Data base.....	5
2.4 Web service container.....	5
3 CONFIGURATION.....	6
3.1 Mail client.....	6
3.2 Binding component SOAP.....	6
3.3 DB table creation.....	7
3.4 XQuare Binding Component.....	7
4 RUN.....	8
5 DOCUMENT MANAGEMENT.....	9

1 INTRODUCTION

This demonstrator implements the use case of a booking service. It allows booking a given number of rooms and optionally a restaurant too. From a client point of view, the booking service appears as a *black box*, taking a number of rooms and a restaurant booking option as input, and returning a booking result with a price and a text-message.

Opening the black box, we see the PETALS ESB and the NovaOrchestra SE orchestrator, which deals in background with hotel and restaurant booking services, mail service, and an SGBD accessed through XQuare.

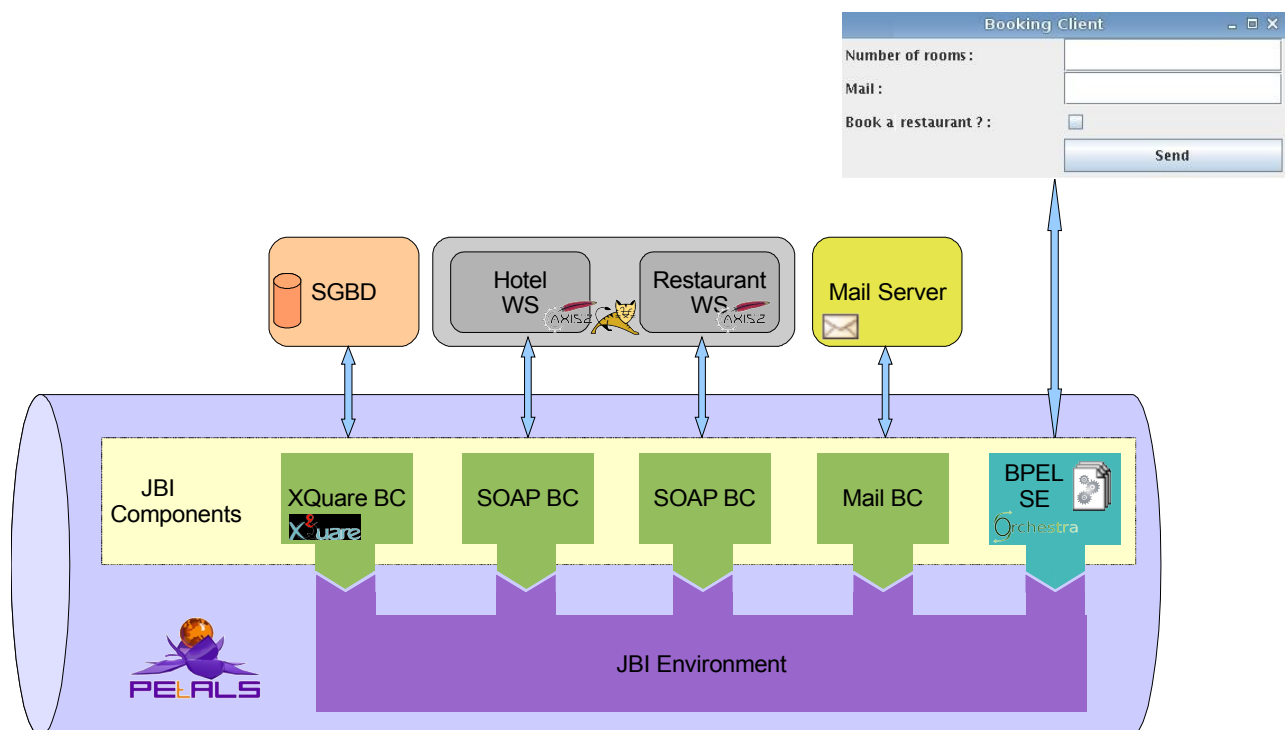


Figure 1 : Demonstrator architecture

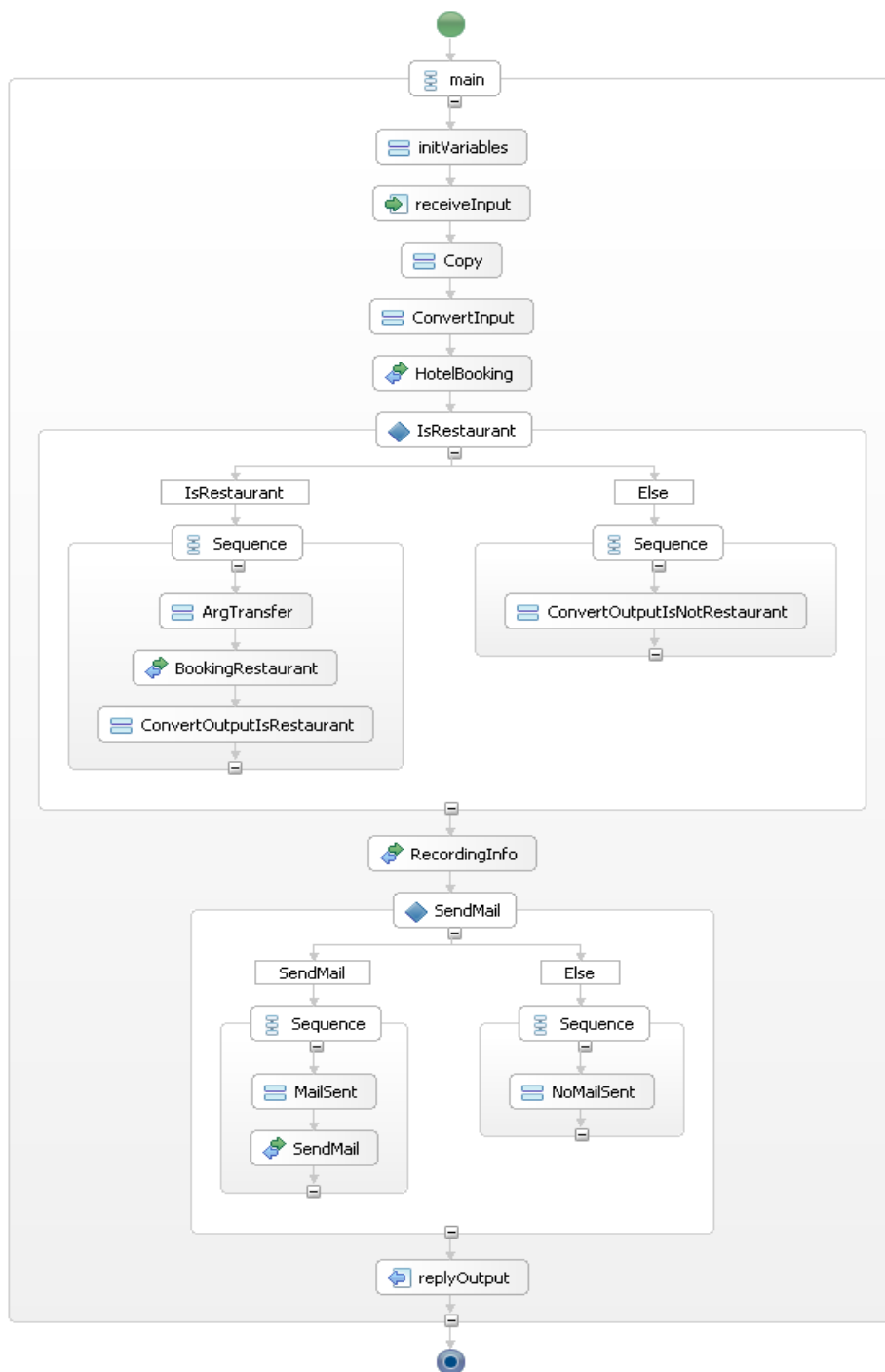


Figure 2 : The orchestration process

2 REQUIREMENTS

2.1 ESB

Name:	PEtALS
Version:	2.1.1.1
Url:	http://forge.objectweb.org/project/download.php?group_id=213&file_id=10089

2.2 Mail Server (optional)

You can use the mail service of your ISP.

Name:	JAMES
Version:	2.3.1
Url:	http://james.apache.org/

2.3 Data base

Name:	MySQL Community Server
Version:	5.0
Url:	http://www.mysql.com/

2.4 Web service container

Name:	Tomcat
Version:	6.0.14
Url:	http://tomcat.apache.org/

3 CONFIGURATION

In the following chapters, you will have to replace blue-text sections with your own parameters.

3.1 Mail client

SendMail.zip\su-sendmail.zip\META-INF\jbi.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<jbi:jbi xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:petals="http://petals.ow2.org/extensions"
  xmlns:jsb="http://petals.ow2.org/"
  xmlns:jbi="http://java.sun.com/xml/ns/jbi"
  version="1.0">
  <jbi:services binding-component="true">
    <jbi:provides interface-name="jsb:SendMail"
      service-name="jsb:MailService"
      endpoint-name="SendMailServiceEndpoint">
      <petals:params>
        <petals:param name="scheme">smtp</petals:param>
        <petals:param name="hostname">smtp.inria.fr</petals:param>
        <petals:param name="username">ravoajan</petals:param>
        <petals:param name="from">
          ravoajan@inrialpes.fr
        </petals:param>
        <petals:param name="to">
          ravoajan@inrialpes.fr
        </petals:param>
      </petals:params>
      <petals:wSDL>/WSDL/MailService.wSDL</petals:wSDL>
    </jbi:provides>
  </jbi:services>
</jbi:jbi>
```

3.2 Binding component SOAP

RestaurantBooking.zip\su-restaurantbooking.zip\META-INF\jbi.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<jbi:jbi version="1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:soap="http://petals.ow2.org/extensions"
  xmlns:jbi="http://java.sun.com/xml/ns/jbi"
  xmlns:wSDL="https://wiki.objectweb.org/ESBi"
  xmlns:petals="http://petals.ow2.org/">
  <jbi:services binding-component="true">
    <jbi:provides interface-name="wSDL:RestaurantBooking"
      service-name="wSDL:RestaurantBookingService"
      endpoint-name="RestaurantBookingServiceEndpoint">
      <soap:wSDL>
        http://XXX/restaurantbooking?wSDL
      </soap:wSDL>
      <soap:params>
        <soap:param name="address">
          http://XXX/restaurantbooking
        </soap:param>
      </soap:params>
    </jbi:provides>
  </jbi:services>
</jbi:jbi>
```

HotelBooking.zip\su-hotelbooking.zip\META-INF\jbi.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<jbi:jbi version="1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:soap="http://petals.ow2.org/extensions"
  xmlns:jbi="http://java.sun.com/xml/ns/jbi"
  xmlns:wsdl="https://wiki.objectweb.org/ESBi"
  xmlns:petals="http://petals.ow2.org/">
  <jbi:services binding-component="true">
    <jbi:provides interface-name="wsdl:HotelBooking"
      service-name="wsdl:HotelBookingService"
      endpoint-name="HotelBookingServiceEndpoint">
      <soap:wsdl>
        http://XXX/hotelbooking?wsdl
      </soap:wsdl>
      <soap:params>
        <soap:param name="address">
          http://XXX/hotelbooking
        </soap:param>
      </soap:params>
    </jbi:provides>
  </jbi:services>
</jbi:jbi>
```

3.3 DB table creation

In your DB, you need to create a table “booking” in your schema (called “xquare_test” in our example).

```
create table booking (
  nb_room          integer,
  email            varchar(50),
  restaurantbooked boolean
);
```

3.4 XQuare Binding Component

demo-sa-xquare-1.0.zip\demo-su-nmr-xquare-1.0.zip\META-INF\jbi.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<jbi:jbi xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:petals="http://petals.ow2.org/extensions"
  xmlns:xquare="http://www.xquark.org/Booking"
  xmlns:jbi="http://java.sun.com/xml/ns/jbi" version="1.0">
  <jbi:services binding-component="false">
    <jbi:provides interface-name="xquare:Xquare"
      service-name="xquare:SampleXquareService"
      endpoint-name="XquareEndpoint">
      <!-- Specific parameters for this service unit -->
      <petals:params>
        <petals:param name="xquare.database.url">
          jdbc:mysql://localhost/xquare_test
        </petals:param>
        <petals:param name="xquare.database.user">root</petals:param>
        <petals:param name="xquare.database.password">xxx</petals:param>
        <petals:param name="xquare.database.mapping">
          booking.map
        </petals:param>
      </petals:params>
      <petals:wsdl>xquare.wsdl</petals:wsdl>
    </jbi:provides>
  </jbi:services>
</jbi:jbi>
```

4 RUN

- Launch tomcat
- Copy/Paste the .war located inside [Orchestration_Demo_v3.zip](#) archive into the webapps tomcat's folder.
- Launch PEtALS
- Copy/Paste the artefacts located inside [Orchestration_Demo_v3.zip](#) archive a in "*install*" PEtALS folder in this order:
 - petals-se-orchestra-1.0-beta1.zip
 - petals-bc-mail-3.0-SNAPSHOT.zip
 - petals-bc-soap-3.0.zip
 - petals-bc-xquare-2.0-SNAPSHOT-insert-noreturn.zip
 - petals-sample-client-1.4.3.zip (**optional**)
 - SendMail.zip
 - HotelBooking.zip
 - RestaurantBooking.zip
 - BookingConsume.zip
 - MailConsume.zip
 - demo-sa-xquare-1.0.zip
 - sa-booking-bpel2.zip

Now you can use the demonstrator through the web service exposed by Orchestra SE to this URL: <http://localhost:8084/petals/services/BookingService>

5 DOCUMENT MANAGEMENT

List of Releases:

Release	Date	§ ¹	Action ²	Modification
V0.1	19/03/2008		C	Document creation

¹ § = chapter(s) updated

² Action = C : Creation, U : Update, D : Deletion
