|  |
| --- |
| Software Engineering of Distributed Systems, KTH |
| Programming Web Services Project |
| Flight Ticket Reservation based on Web Service and BPEL |

|  |
| --- |
| Sike Huang and Shanbo Li  2008-3-2 |

## Description of the problem and design solutions

There are several problems to be solved in this project:

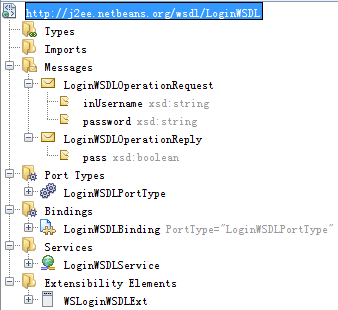
1. Authorization of customers, only credential user can access the services
2. Given a departure city, destination city and date, possible itineraries should be checked.
3. In case there is no direct flight, combination of flights is considered.
4. Cheapest available flight will be issued to customer.

We design the following solution:

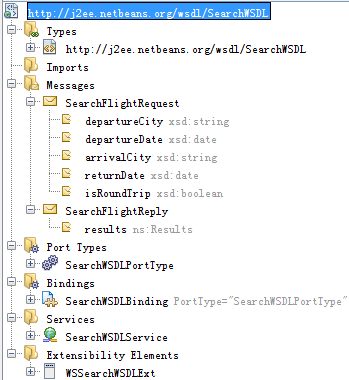


On the application server, there are four web services running, LoginWS takes care of authentication, SearchWS finds out the cheapest flight, and it calls PaypalWS and ConfirmWS to charge money for credit card and confirm the flight again available seats with in an atomic transaction, the business process is managed by a workflow in BPM, and all services rely on EJB / JPA to access the MySQL database.

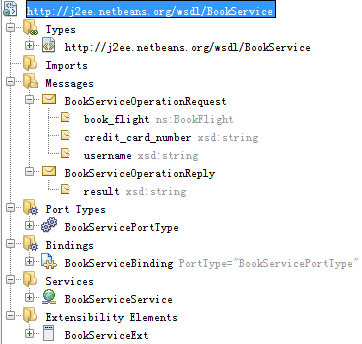
## WSDL description of all implemented services



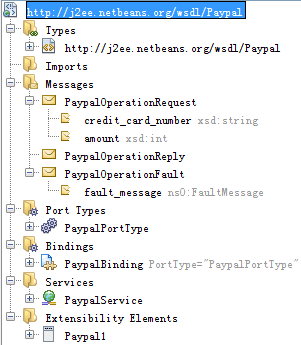
|  |  |
| --- | --- |
| LoginWSDL.wsdl | |
| Authentication of customer again username and password stored in database | |
| Request: | **Response:** |
| Username  Password | Correct username and password or not |



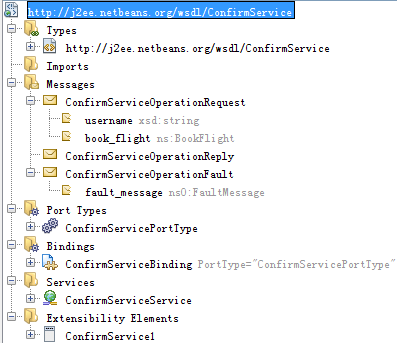
|  |  |
| --- | --- |
| SearchWSDL.wsdl | |
| Finds out possible itineraries, and select the cheapest flights | |
| Request: | **Response:** |
| Departure city  Departure date  Arrival city  Return date (if round trip)  Round trip or one way | Cheapest combination of flights |



|  |  |
| --- | --- |
| BookService.wsdl | |
| Book flights (obtained from SearchWS) by calling ConfirmWS and PaypalWS in a WS-AT | |
| Request: | **Response:** |
| Flights to be booked  Credit card number | System message (whether succeed in booking or not) |



|  |  |
| --- | --- |
| PaypalService.wsdl | |
| Charge certain amount of money from a credit card | |
| Request: | **Fault:** |
| Credit card number  Amount to be charged | Invalid credit card number, or insufficient money (If failed in charging) |



|  |  |
| --- | --- |
| ConfirmService.wsdl | |
| Confirm the seat in a flight if available (reduce availability afterwards) | |
| Request: | **Fault:** |
| Flights to be confirmed | No available seat, failed in confirming |

## Java source code (or BPEL code) for all implemented services

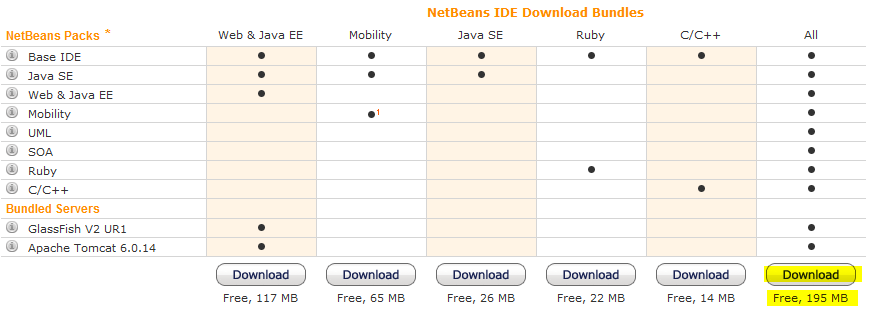
See **SearchFlightBPEL.bpel** under **FlightBpelModule** for BPEL code.

See **FlightEnterpriseApplication** and **FlightHelperEnterpriseApplication** for Java source code.

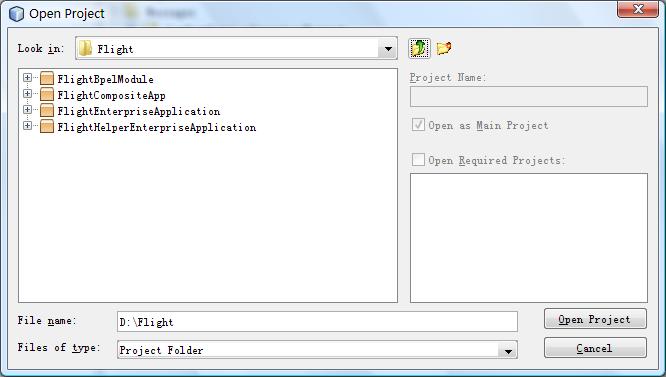
## Protocols of services deployment

Download and install Netbeans IDE 6.0.1 (Packs All)

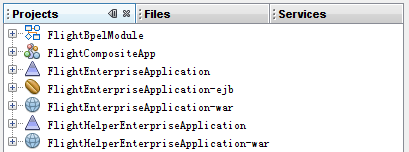
From <http://download.netbeans.org/netbeans/6.0/final/>



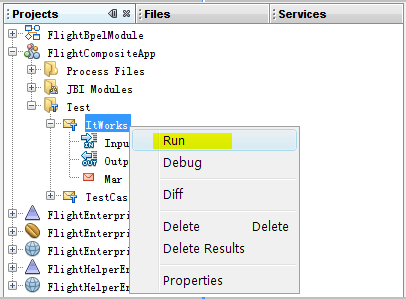
Import the project (four modules)



You shall see the following in your NetBeans workspace, and you should deploy **FlightHelperEnterpriseApplication** first, then deploy **FlightEnterpriseApplication**, and last deploy **FlightCompositeApp**.



Now you are ready to run the test case.



## Texts of all constructed incoming and outgoing SOAP messages for invoking/replying all implemented services.

**Invoke:**

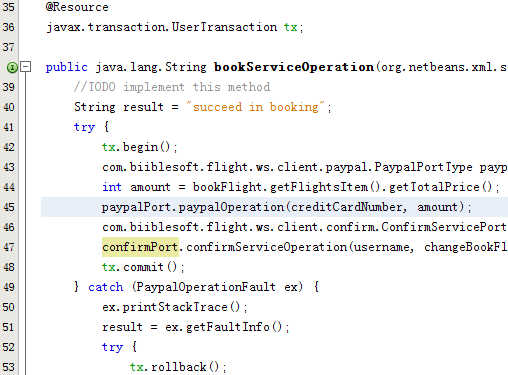
http://www.blogjava.net/Images/OutliningIndicators/None.gif<soapenv:Envelope xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:cli="http://j2ee.netbeans.org/wsdl/ClientSearch">  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  <soapenv:Body>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif    <cli:ClientSearchOperation>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif      <username>sikeh</username>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif      <password>sikeh</password>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif      <departureCity>Stockholm</departureCity>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif      <departureDate>2008-06-24</departureDate>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif      <arrivalCity>Sydney</arrivalCity>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif      <returnDate>2999-12-30</returnDate>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif      <isRoundTrip>true</isRoundTrip>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif      <creditCardNumber>cn123456</creditCardNumber>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif    </cli:ClientSearchOperation>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  </soapenv:Body>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif</soapenv:Envelope>

**Reply:**

http://www.blogjava.net/Images/OutliningIndicators/None.gif<?xml version="1.0" encoding="UTF-8"?>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  <SOAP-ENV:Body>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif    <m:ClientSearchOperationResponse xmlns:m="http://j2ee.netbeans.org/wsdl/ClientSearch">  
http://www.blogjava.net/Images/OutliningIndicators/None.gif      <result xmlns="">  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        <ns0:CheapestFlightItem xmlns:ns0="http://xml.netbeans.org/schema/FlightSchema">  
http://www.blogjava.net/Images/OutliningIndicators/None.gif          <ns0:GoFlights>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif            <ns0:GoFlight1>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:FlightCode xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">CA912</ns0:FlightCode>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:DepartureCity xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">Stockholm</ns0:DepartureCity>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:DepartureTime xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">05:50:00.000+08:00</ns0:DepartureTime>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:ArrialCity xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">Beijing</ns0:ArrialCity>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:ArrialTime xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">14:50:00.000+08:00</ns0:ArrialTime>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:Price xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">1124</ns0:Price>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif            </ns0:GoFlight1>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif            <ns0:NumOfGoFlights xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">2</ns0:NumOfGoFlights>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif            <ns0:Price xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">9722</ns0:Price>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif            <ns0:GoFlight2>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:FlightCode xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">XX98</ns0:FlightCode>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:DepartureCity xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">Beijing</ns0:DepartureCity>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:DepartureTime xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">15:00:00.000+08:00</ns0:DepartureTime>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:ArrialCity xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">Sydney</ns0:ArrialCity>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:ArrialTime xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">20:00:00.000+08:00</ns0:ArrialTime>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:Price xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">8598</ns0:Price>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif            </ns0:GoFlight2>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif          </ns0:GoFlights>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif          <ns0:TotalPrice xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">10537</ns0:TotalPrice>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif          <ns0:NumOfGoFlights xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">2</ns0:NumOfGoFlights>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif          <ns0:BackFlights>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif            <ns0:BackFlight1>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:FlightCode xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">HSK416</ns0:FlightCode>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:DepartureCity xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">Sydney</ns0:DepartureCity>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:DepartureTime xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">01:01:01.000+08:00</ns0:DepartureTime>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:ArrialCity xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">Beijing</ns0:ArrialCity>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:ArrialTime xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">02:02:02.000+08:00</ns0:ArrialTime>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:Price xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">5</ns0:Price>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif            </ns0:BackFlight1>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif            <ns0:NumOfBackFlights xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">2</ns0:NumOfBackFlights>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif            <ns0:Price xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">815</ns0:Price>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif            <ns0:BackFlight2>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:FlightCode xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">CA911</ns0:FlightCode>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:DepartureCity xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">Beijing</ns0:DepartureCity>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:DepartureTime xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">14:25:00.000+08:00</ns0:DepartureTime>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:ArrialCity xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">Stockholm</ns0:ArrialCity>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:ArrialTime xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">23:59:00.000+08:00</ns0:ArrialTime>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif              <ns0:Price xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">810</ns0:Price>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif            </ns0:BackFlight2>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif          </ns0:BackFlights>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif          <ns0:NumOfBackFlights xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL">2</ns0:NumOfBackFlights>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        </ns0:CheapestFlightItem>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        <ns0:SearchSuccessful xmlns:msgns="http://j2ee.netbeans.org/wsdl/SearchWSDL" xmlns:ns0="http://xml.netbeans.org/schema/FlightSchema">true</ns0:SearchSuccessful>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        <ns0:LoginSuccessful xmlns:ns0="http://xml.netbeans.org/schema/FlightSchema">true</ns0:LoginSuccessful>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        <ns0:ReturnMessage xmlns:msgns="http://j2ee.netbeans.org/wsdl/BookService" xmlns:ns0="http://xml.netbeans.org/schema/FlightSchema">no available seat, failed in booking!</ns0:ReturnMessage>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif        <ns0:BookSuccessful xmlns:ns0="http://xml.netbeans.org/schema/FlightSchema">false</ns0:BookSuccessful>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif      </result>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif    </m:ClientSearchOperationResponse>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif  </SOAP-ENV:Body>  
http://www.blogjava.net/Images/OutliningIndicators/None.gif</SOAP-ENV:Envelope>

## Explanation of your solution for implementing transactional control in the project

Project Tango creates an Atomic Transaction Context the first time a transacted Web service operation is invoked within a JTA transaction scope. For example, in the code fragment shown in Code Sample below, WS-AT transaction context is created in line 45:



This does not impose any overhead on JavaEE transactions when the capability is not being used. This context flows along with the application messages. A request from a .NET 3.0 client that contains transaction context is also understood by Project Tango and participates in the transactional context.

The Transactions feature in Tango supports the following:

● WS-AT Transactions flowing from the Web tier and EJB tier

● Automated mapping of Container Managed Transactions to semantically equivalent WSAtomic

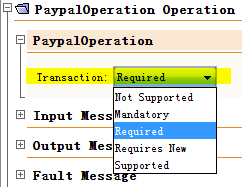
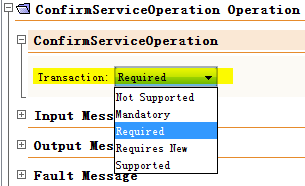
Transaction policy assertions

● Auto-enlistment of XAResources – JMS implementation in GlassFish V2, Entity manager of

TopLink Essentials using JavaDB, Entity Manager of Hibernate

The WS-AT specification also defines policy assertions to describe transactional capabilities of an

endpoint. Transactional capabilities on a Web service endpoint can be easily enabled on a per-operation basis using NetBeans IDE. Figure below shows the different options available to configure WS-AT support on an operation. Selecting a Transaction type on an operation generates the appropriate WS-AT policy assertion in the WSIT configuration file. This configuration file is then used to generate the WSDL. The WSIT tutorial has more details.



The transactions feature of Project Tango works only with the GlassFish V2 container.

## Analysis of advantages and drawbacks of your solution, any remarks about implementing and usage of Web services.

Our solution is module-based, hence flexible to make any changes, and it fully utilizes Java EE technologies, such as Java Persistence API, Enterprise JavaBean and etc.

However there is still limitation on the itineraries look-up algorithm, so far, we have direct one-way/roud-trip flight and one-stop one-way/round-trip flight, it shall be possible to combine more than two flights in either way.

Studies of WS-\* and BPEL is a very interesting yet challenging work, but it is hard to find implementation of those specification and tutorials about how to use, so far as I know, only Axis1/2, Metro@Glassfish have fulfilled WS-Coordination and WS-AtomicTransaction (ref: <http://wiki.apache.org/ws/StackComparison>), and it seems only Apache Muse supports WS-Resource, so we drop it.