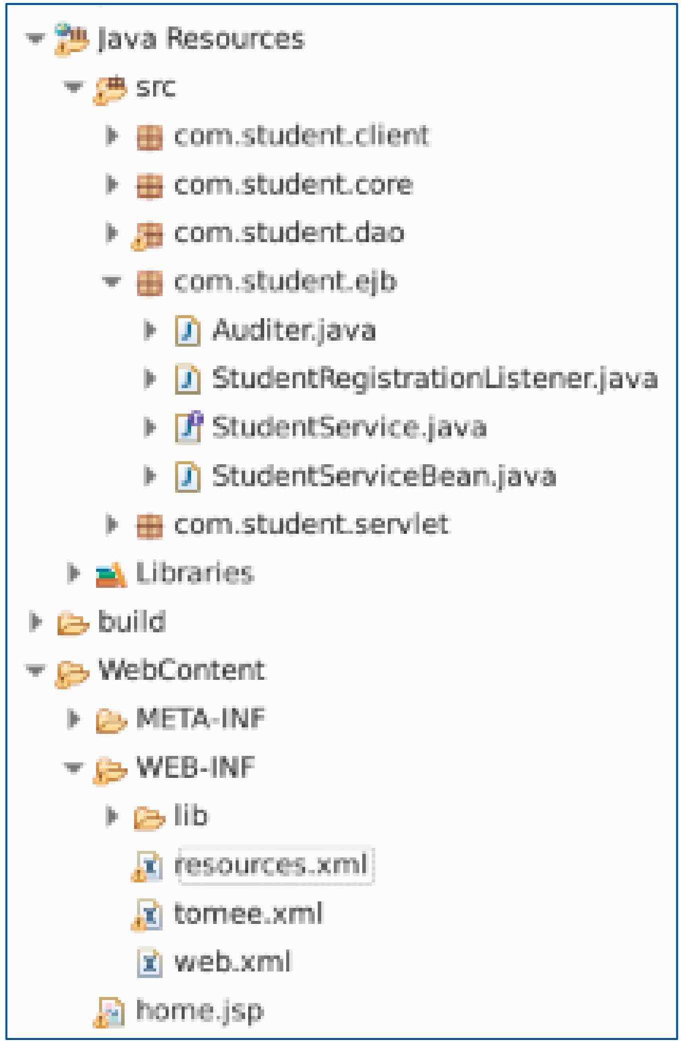
Lab 5.1 – Message Driven Beans

- We will notify a Listener asynchronously when a single student is requested via our StudentServlet

- From your Lab setup directory for this lab;
  - ❑ Copy the tomee.xml file under your WEB-INF directory. This defines the Message Broker, the JMS Container, the JDNI administered ConnectionFactory and Destination
  - ❑ Create a class StudentRegistrationListener in package com.student.ejb
  - ❑ It is to implement MessageListener
  - ❑ Subsequently we will have to implement the method public void onMessage(Message message)



```
Java Resources
  src
    com.student.client
    com.student.core
    com.student.dao
    com.student.ejb
      Auditer.java
      StudentRegistrationListener.java
      StudentService.java
      StudentServiceBean.java
    com.student.servlet
  Libraries
build
WebContent
  META-INF
  WEB-INF
    lib
    resources.xml
    tomee.xml
    web.xml
  home.jsp
```

Using resource.xml

Defines the JMS Container, JMS Broker and JNDI Administered Objects

```
<tomee>
    <Resource id="MyJmsResourceAdapter" type="ActiveMQResourceAdapter">
     BrokerXmlConfig  = (tcp://localhost:61616)
     ServerUrl = tcp://localhost:61616
  </Resource>

  <Resource id="MyJmsConnectionFactory" type="javax.jms.ConnectionFactory">
     ResourceAdapter = MyJmsResourceAdapter
  </Resource>

  <Container id="MyJmsMdbContainer" ctype="MESSAGE">
     ResourceAdapter = MyJmsResourceAdapter
  </Container>

  <Resource id="FooQueue" type="javax.jms.Queue"/>
</tomee>
```

Connectionfactory ties to embedded JMS Broker in Tomee

Remember the name FooQueue and it's a Queue

The Consumer

- Annotate the class itself as MessageDriven
- Add TWO ActivationConfigProperties as below where we define the destination type and name. This is from the tomee.xml
- This meta data is important for the JMS Container embedded in Tomee

```
@MessageDriven(activationConfig = {
   @ActivationConfigProperty( propertyName = "destinationType",
                       propertyValue = "javax.jms.Queue"),
    @ActivationConfigProperty( propertyName = "destination",
                       propertyValue = "FooQueue")
})
public class StudentRegistrationListener implements MessageListener
```

The Producer

- Our Message Producer is StudentServlet. Inject the Connectionfactory and Destination using @Resource It’s the same Queue as the Consumer

```
@Resource
private ConnectionFactory connectionFactory;
@Resource(name = "FooQueue")
private Queue fooQueue;
```

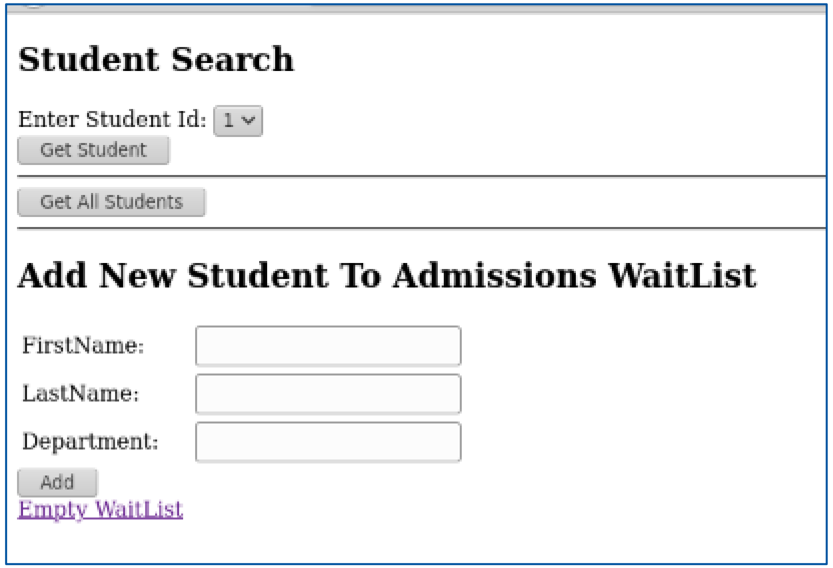
- Create a private method sendMessage((Student student) in the servlet. From your lab setup directory for this lab, you will find the verbose JMS producer code in the file *(producer.txt) that you can cut and paste into the body of this new method

- In your doGet method, call sendMessage(Student student) where you retrieve a single student (not get ALL Students)

- Restart your server and launch your application by selecting the root node of your project->right click->Run on Server

Trigger the Message Driven Bean

- Select a single Student to retrieve



Student Search

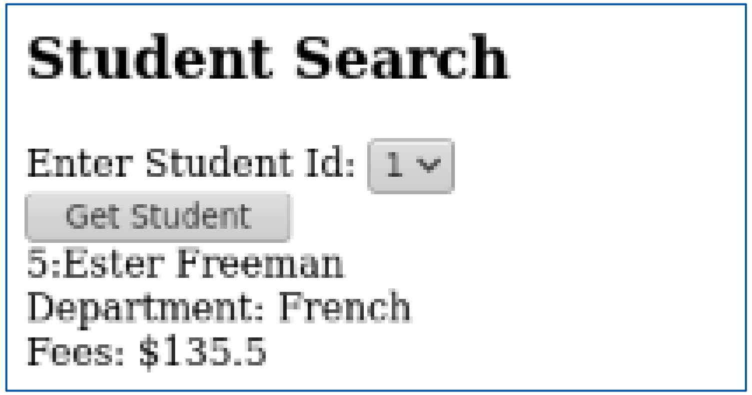
Enter Student Id: 1

Add New Student To Admissions WaitList

FirstName:

LastName:

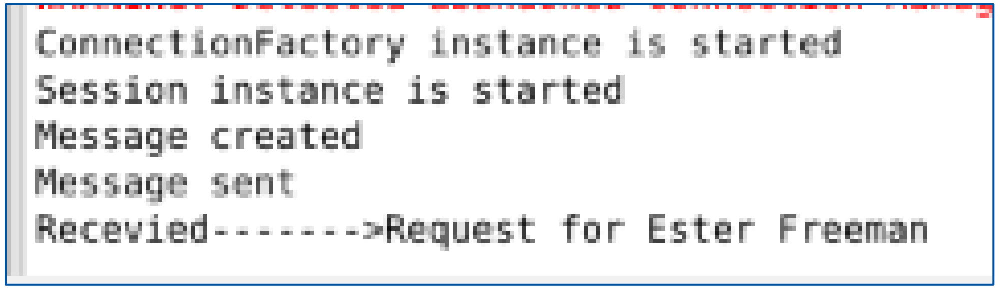
Department:



Student Search

Enter Student Id: 1

5:Ester Freeman
Department: French
Fees: $135.5



```
ConnectionFactory instance is started
Session instance is started
Message created
Message sent
Recevied-------->Request for Ester Freeman
```

- This should trigger the JMS Producer method and the Listener should pick it up via the configured destination (Queue)

- Your Console will confirm this