Лабораторная работа №4
Обмен сообщениями и Java Message Service

Цель работы: получить навыки работы с системами обмена сообщениями и реализации приложений на платформе Java, взаимодействующих путем обмена сообщениями.

Задание

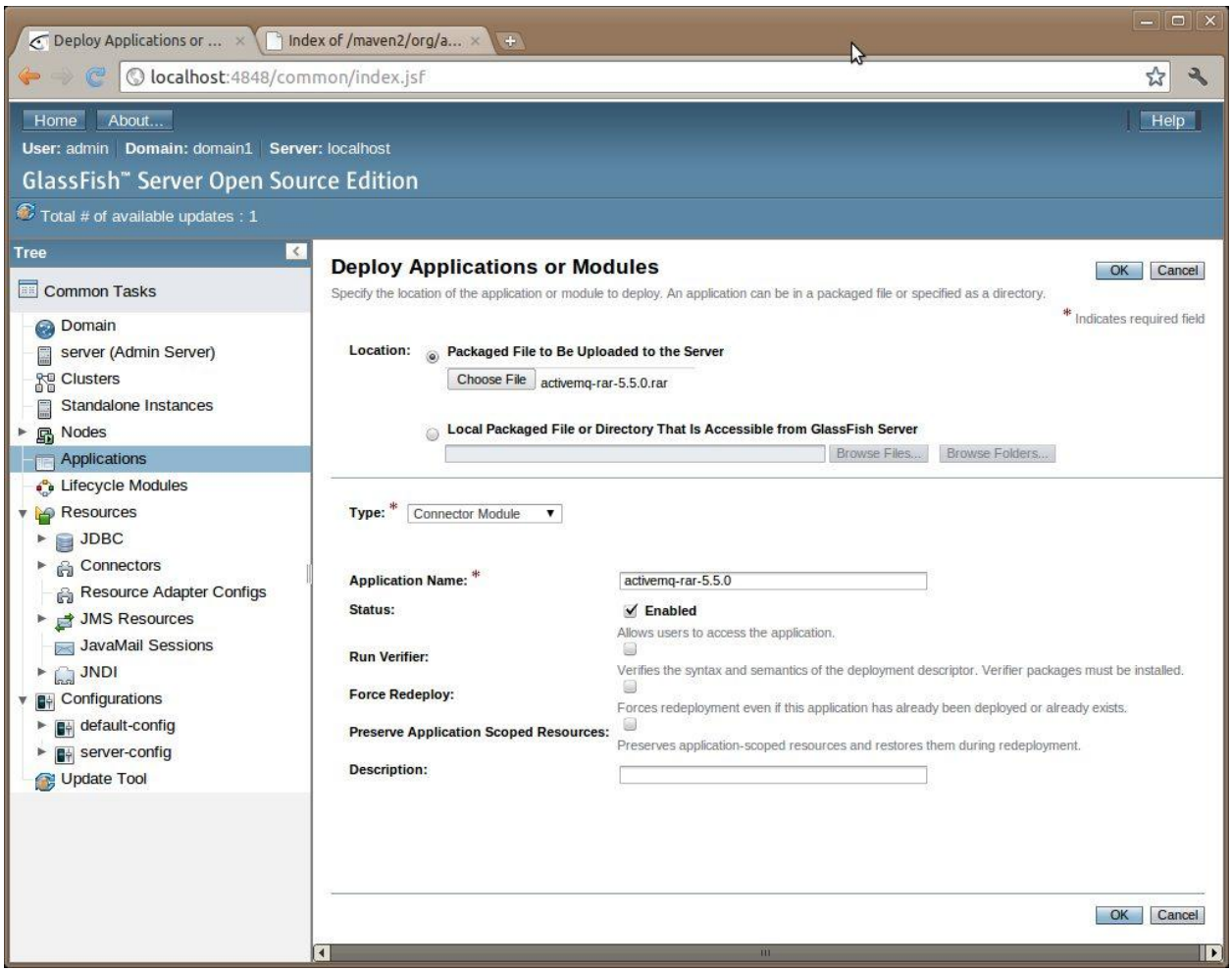
1) Разработать компонент, управляемый сообщениями, получающий сообщения из очереди.
2) Компонент, управляемый сообщениями, должен обрабатывать сообщения-запросы, содержащие Java-объекты.
3) Компонент, управляемый сообщениями, должен взаимодействовать с СУБД (возможно, посредством другого сессионного компонента).
4) Если в сообщении-запросе указан адресат для ответа, то компонент, управляемый сообщениями, должен отправлять указанному адресату ответное текстовое сообщение.
5) Разработать веб-приложение, отправляющее сообщение-запрос в очередь.
6) Разработать клиентское приложение (Java SE), отправляющее сообщение-запрос в очередь и получающее ответ через временную очередь.

Необходимое ПО

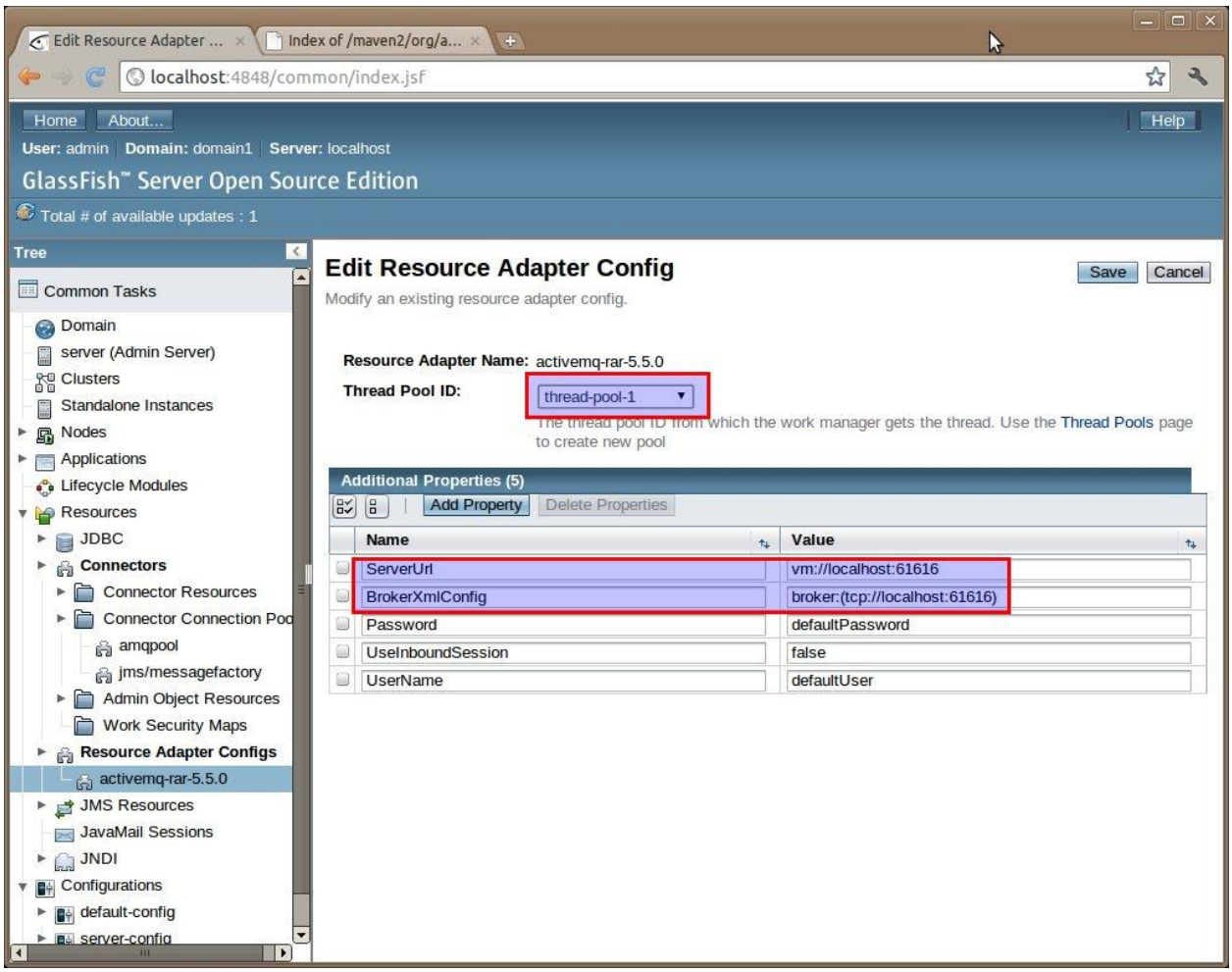
1) NetBeans 7.0.1
2) Glassfish 3.1
3) ActiveMQ 5.5.1 (http://activemq.apache.org/download.html), адаптер ресурсов для Java EE-совместимых серверов приложений и консоль администрирования (см. ниже)

Ход выполнения работы

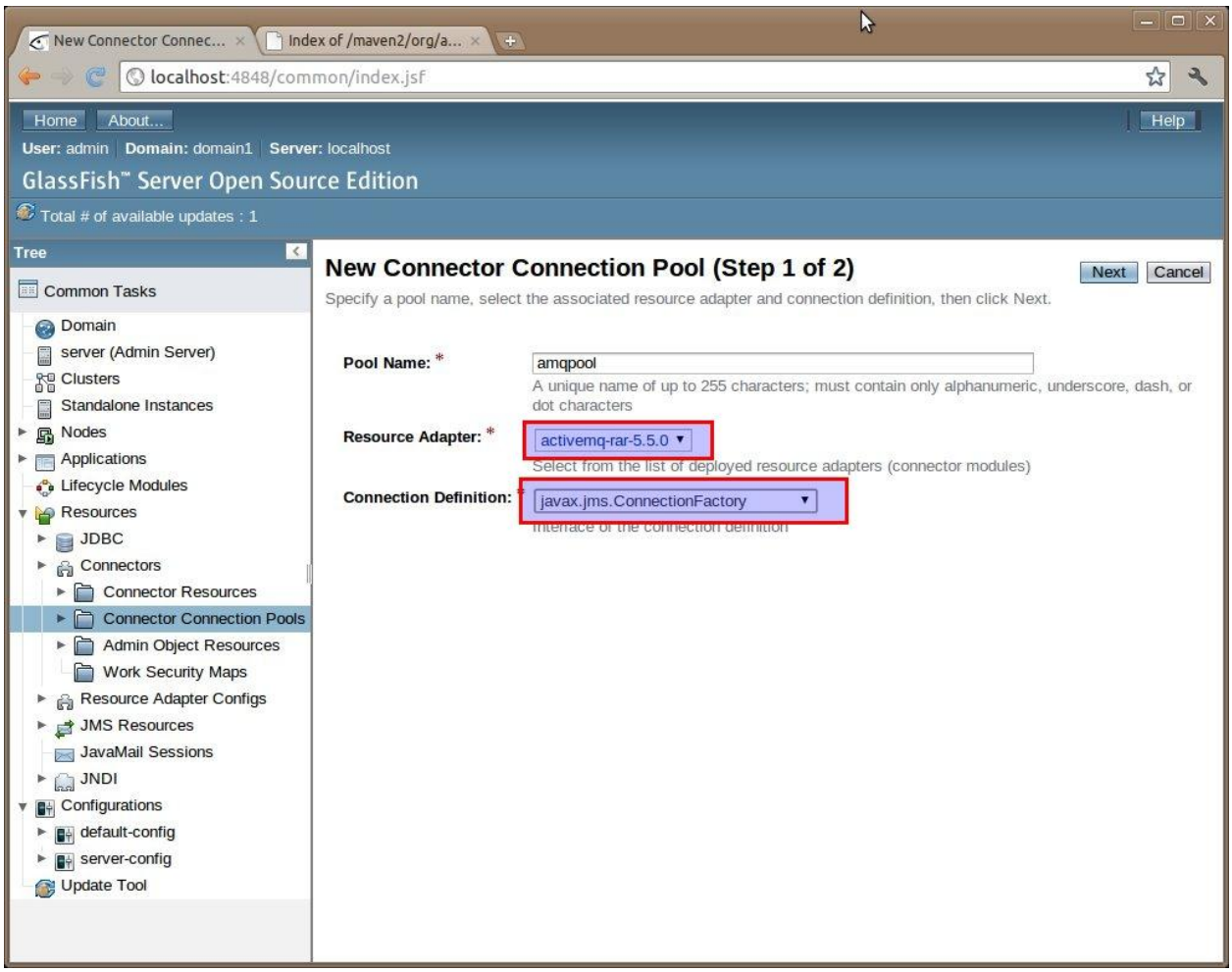
1) Установите систему обмена сообщений ActiveMQ в режиме интеграции с сервером приложений Glassfish
    a. Скопируйте в папку библиотек Glassfish (GLASSFISH_HOME/glassfish/lib) следующие библиотеки, которые находятся в загруженном пакете ActiveMQ:
        i. activemq-all-5.5.1.jar
        ii. slf4j-api-1.5.11.jar
        iii. log4j-1.2.14.jar
        iv. slf4j-log4j12-1.5.11.jar
    b. Загрузите адаптер ресурсов ActiveMQ для Java EE-совместимых серверов приложений (http://repo1.maven.org/maven2/org/apache/activemq/activemq-rar/, файл activemq-rar-5.5.1.rar)
    c. Скопируйте файл activemq-ra-5.5.0.jar из архива activemq-rar-5.5.1.rar в папку библиотек Glassfish (GLASSFISH_HOME/glassfish/lib) и перезагрузите сервер Glassfish, если он запущен.
    d. Откройте Консоль администрирования Glassfish и установите адаптер ресурсов activemq-rar-5.5.1.rar.



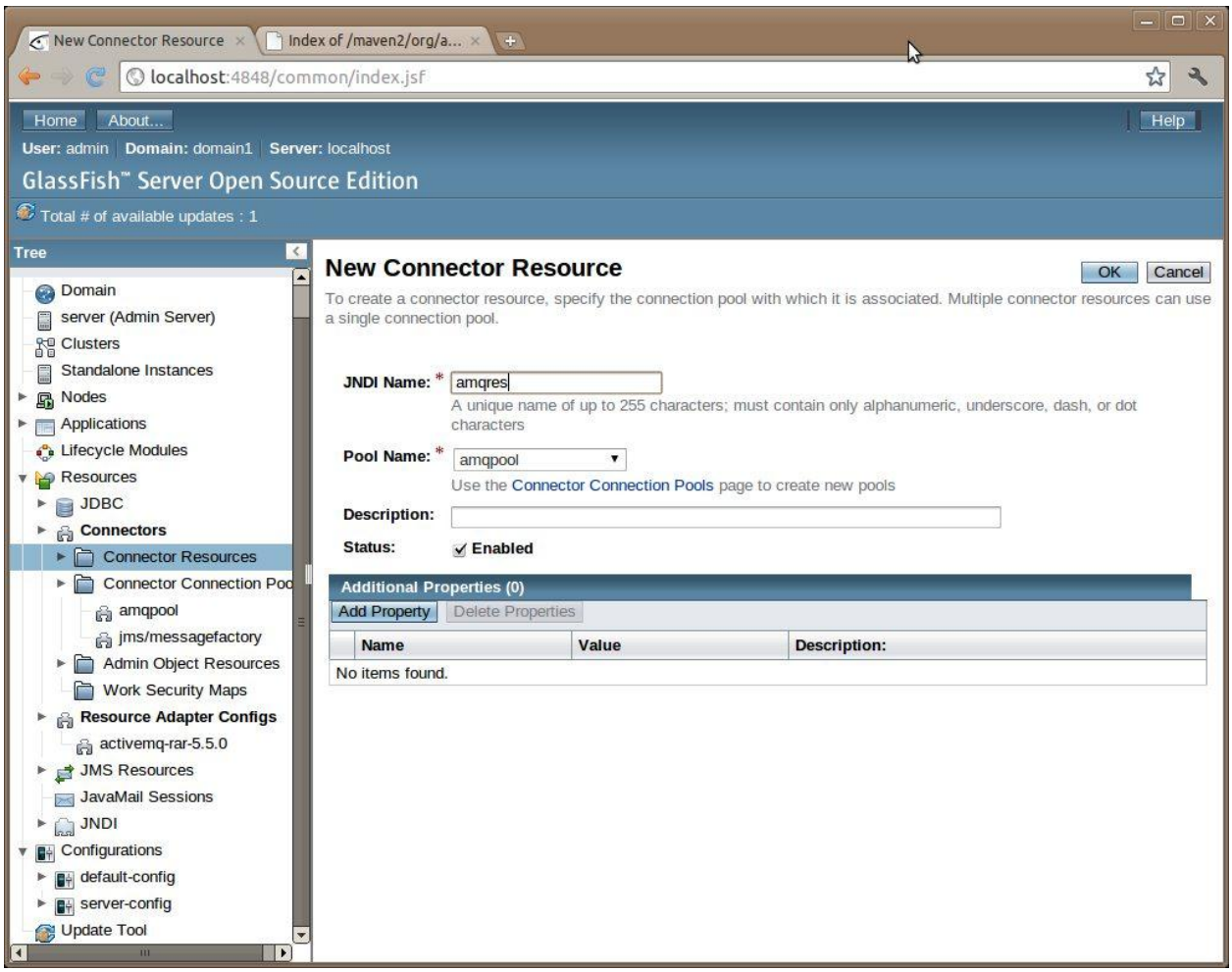
e. Создайте конфигурацию адаптера ресурсов (Resource Adapter Config), выберите пул потоков и установите следующие значения свойств:
    i. ServerUrl: vm://localhost:61616
    ii. BrokerXmlConfig: broker:(tcp://localhost:61616)



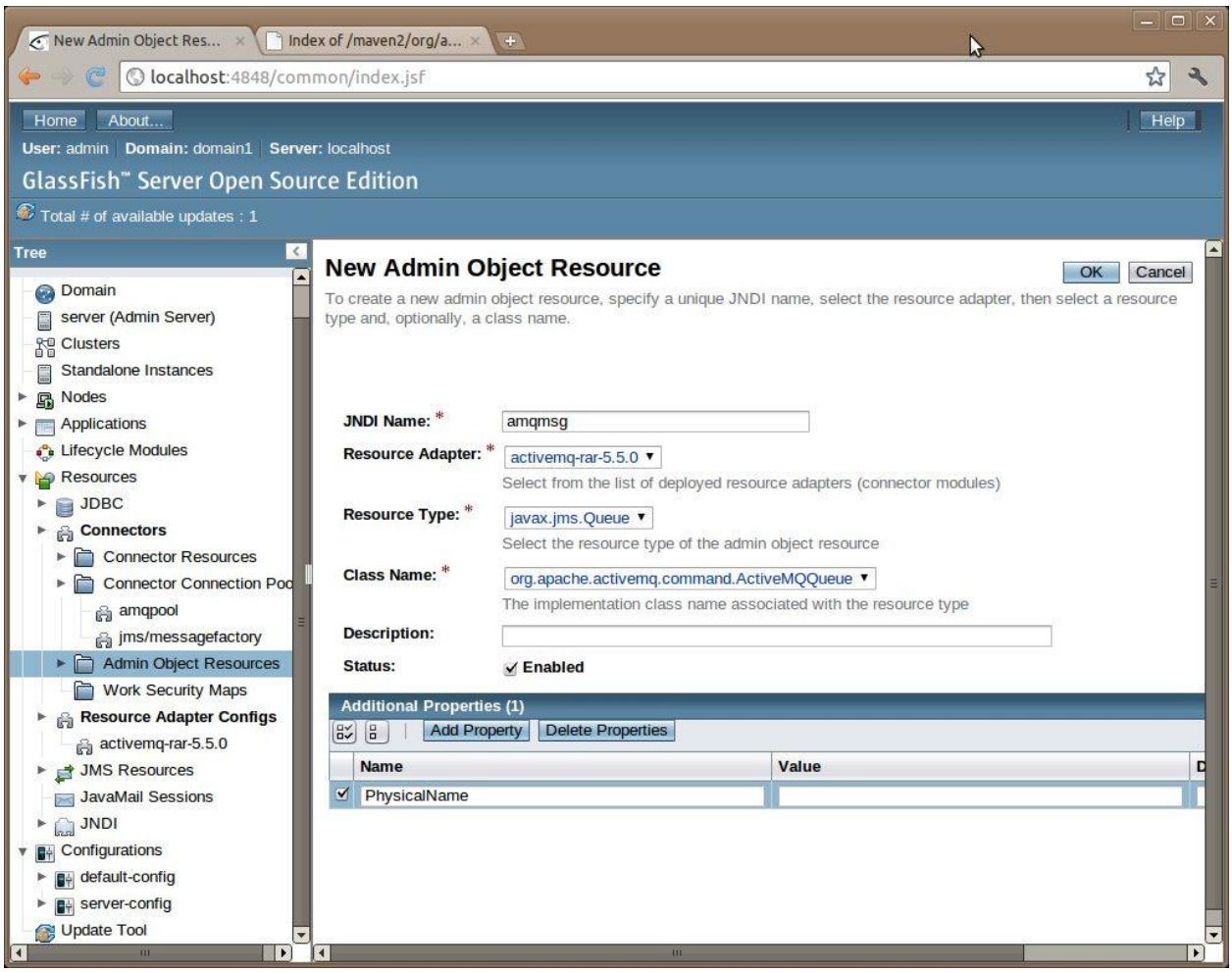
f. Добавьте пул соединений с коннектором (Connector Connection Pool), выберите адаптер ресурсов для ActiveMQ, на второй странице мастера оставьте значения свойств по умолчанию.



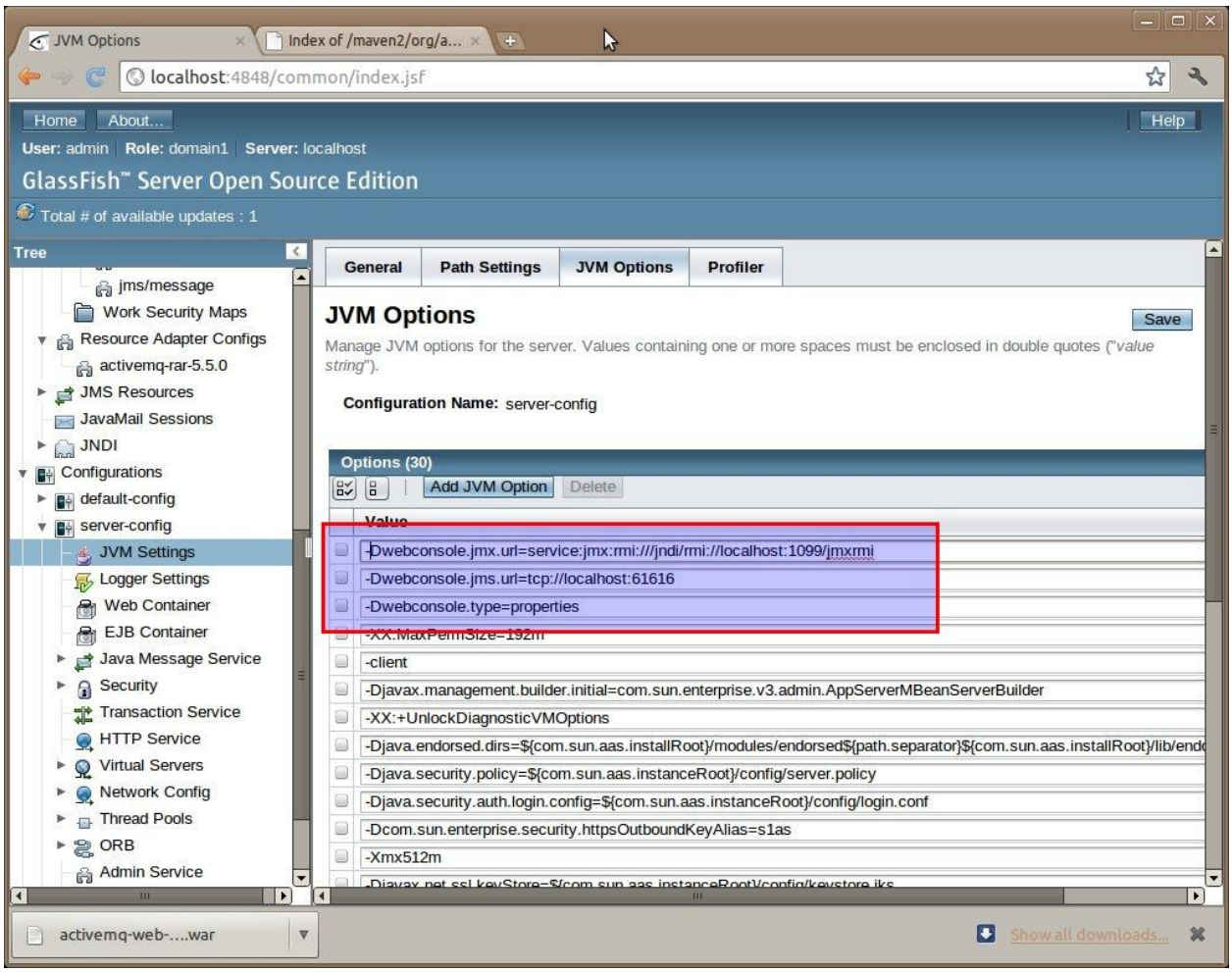
g. Добавьте ресурс коннектора (Connector Resource), укажите в нем созданный на предыдущем шаге пул соединений с коннектором. Ресурс коннектора используется для инъекции фабрики соединений с системой обмена сообщениями.



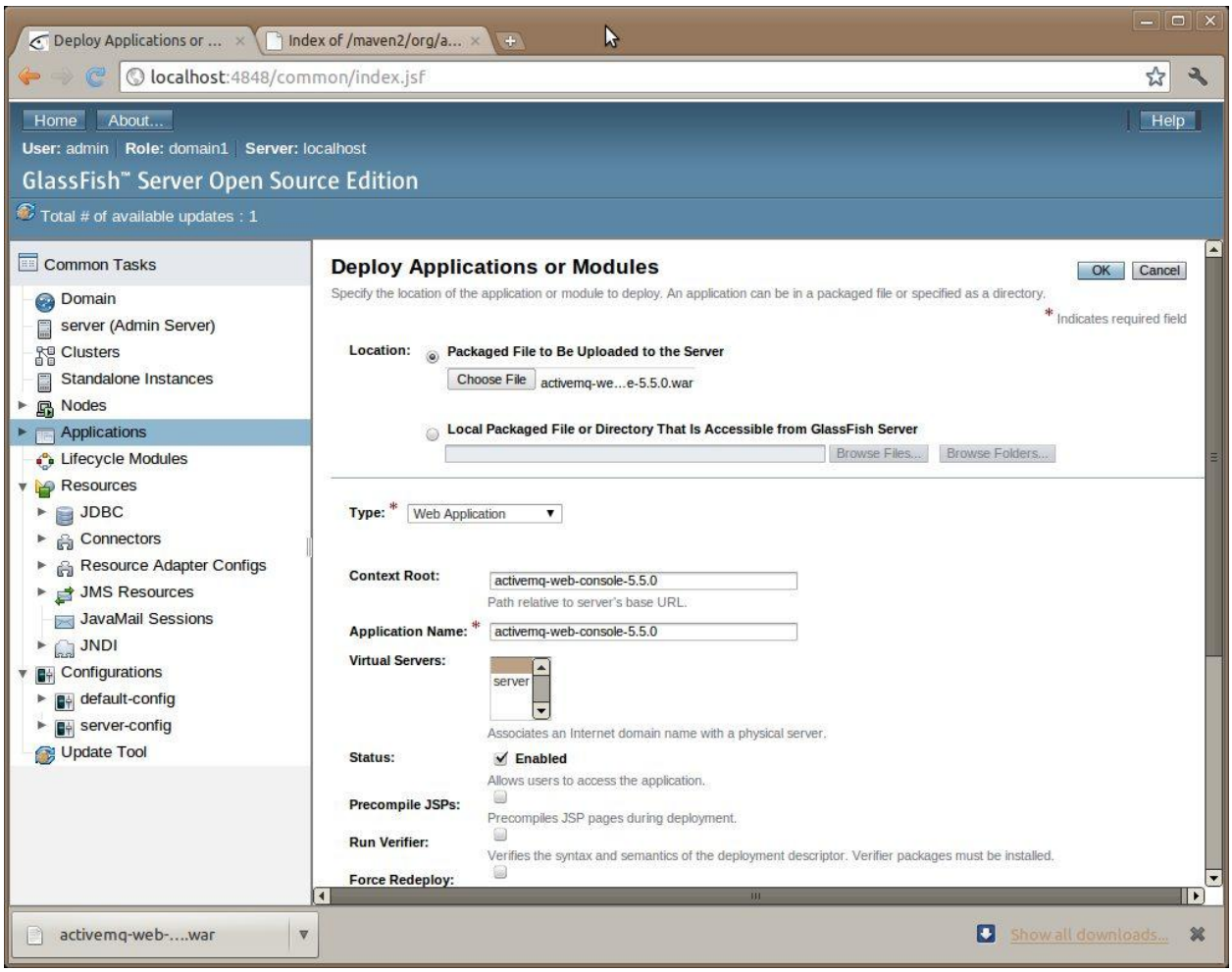
h. Добавьте ресурс администрируемого объекта (Admin Object Resource), выберите для него адаптер ресурса ActiveMQ и тип ресурса «очередь» (Queue).



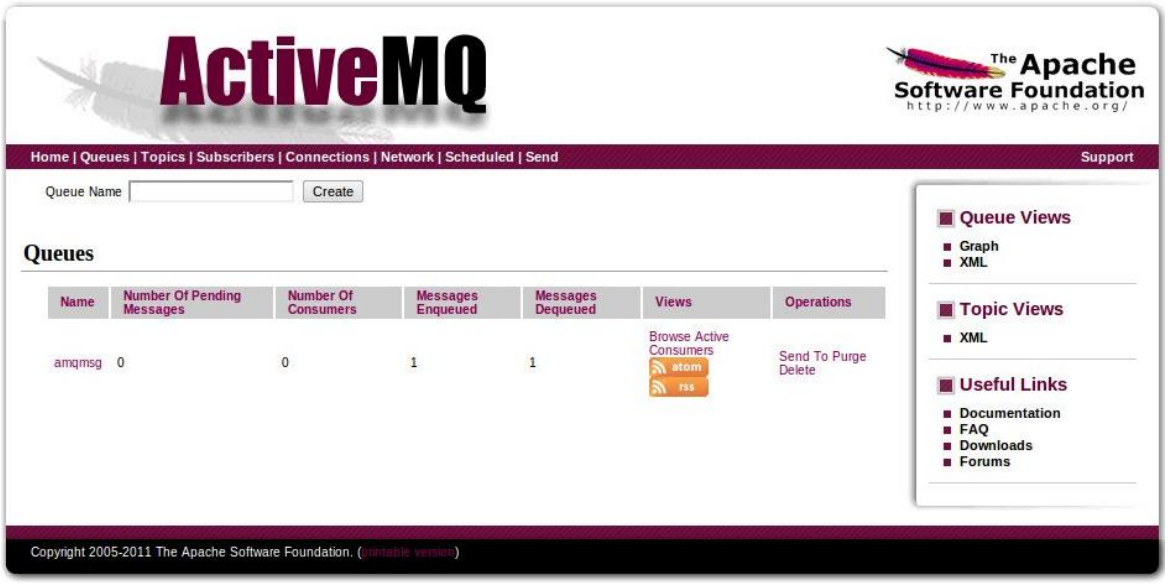
i. Добавьте следующие настройки JVM для запуска сервера приложений (и для default-config, и для server-config) и перезапустите Glassfish:
    i. -Dwebconsole.type=properties
    ii. -Dwebconsole.jms.url=tcp://localhost:61616
    iii. -Dwebconsole.jmx.url=service:jmx:rmi:///jndi/rmi://localhost:1099/jmxrmi



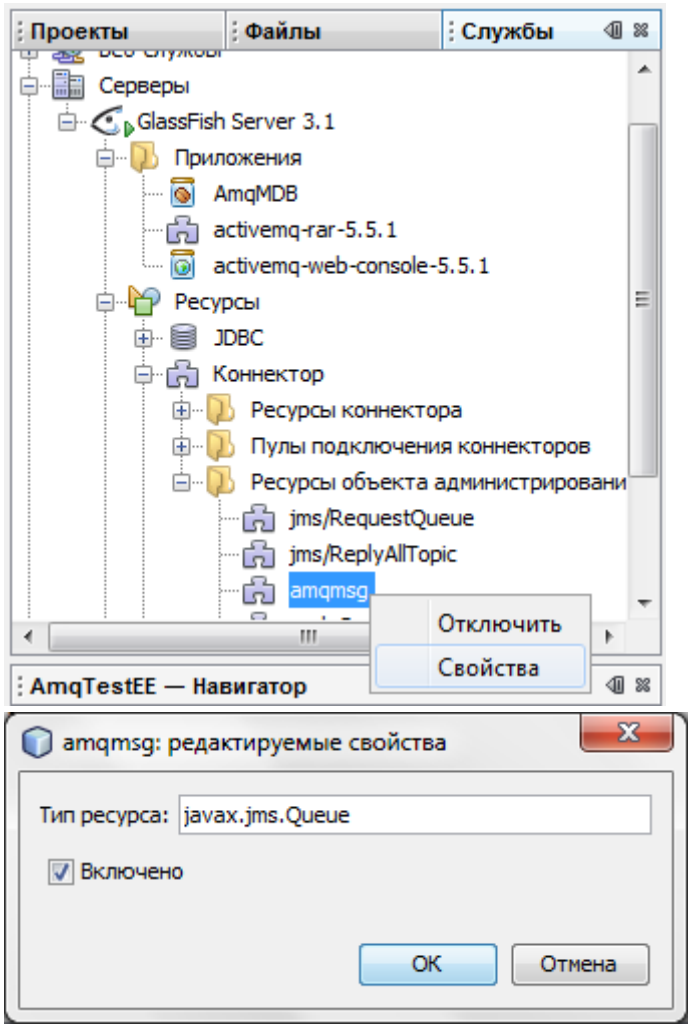
j. Установите веб-приложение консоли администрирования ActiveMQ (загрузка по адресу http://repo1.maven.org/maven2/org/apache/activemq/activemq-web-console/, файл activemq-web-console-5.5.1.war).



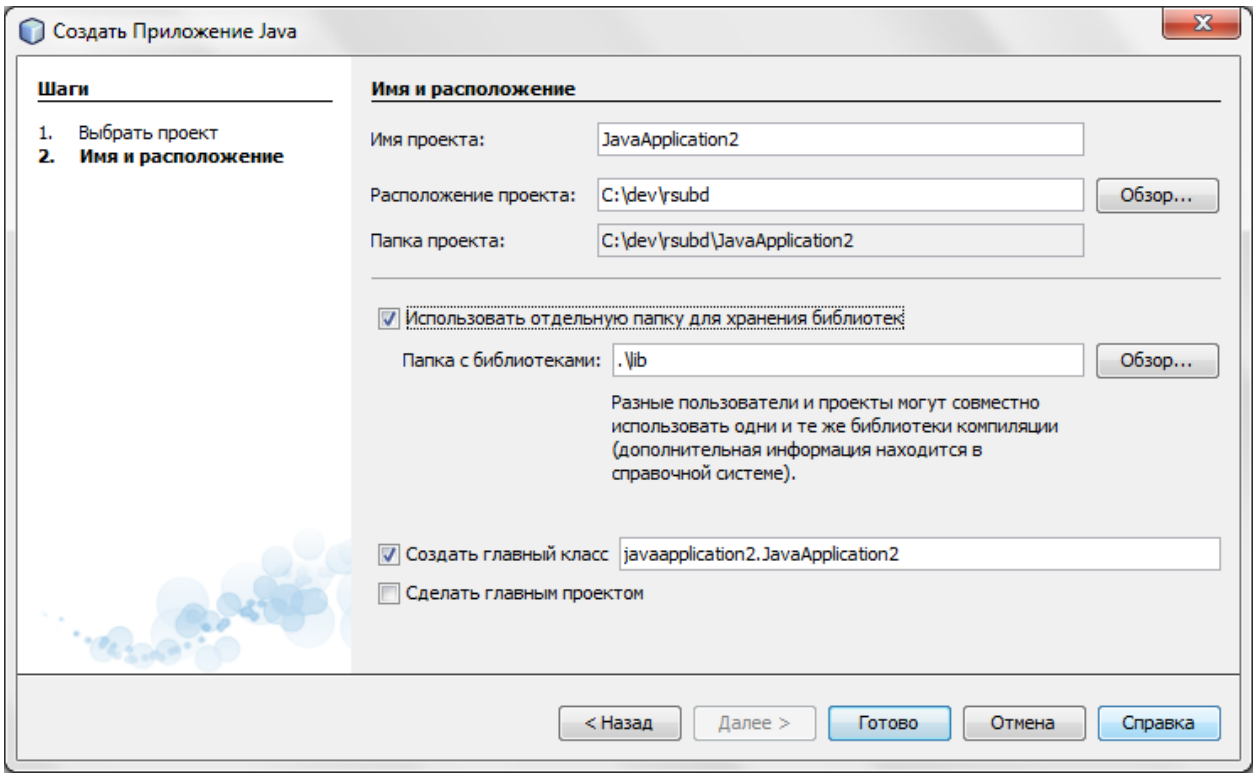
k. Запустите консоль администрирования (http://localhost:8080/activemq-web-console-5.5.1) и создайте очередь с таким же названием, что и созданный ранее ресурс администрируемого объекта.



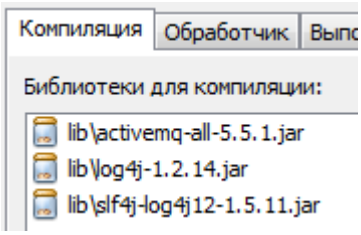
l. Включите созданную ранее очередь в NetBeans. Для этого найдите ее на вкладке Службы (Серверы / GlassFish Server 3.1 / Ресурсы / Коннектор / Ресурсы объекта администрирования), в контекстном меню выберите пункт Свойства и в диалоге свойств установите флажок Включено.



2) Создайте проект клиентского приложения, которое позволит убедиться в правильной настройке очереди.
    a. Создайте проект типа «Java / Приложение Java» со следующими настройками:



b. Добавьте в библиотеки проекта следующие архивы из дистрибутива ActiveMQ.



c. Вставьте следующий код в главный класс приложения:

```
  public static void main(String[] args) {
    try {
      Connection connection = null;
      ActiveMQConnectionFactory connectionFactory = new
ActiveMQConnectionFactory(ActiveMQConnection.DEFAULT_USER,
ActiveMQConnection.DEFAULT_PASSWORD, ActiveMQConnection.DEFAULT_BROKER_URL);
      connection = connectionFactory.createConnection();
      connection.start();
      Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
      Destination destination = session.createQueue(<название_очереди>);
      MessageProducer producer = session.createProducer(destination);
      producer.setDeliveryMode(DeliveryMode.NON_PERSISTENT);

      TextMessage message = session.createTextMessage("THIS IS THE TEST MESSAGE !");
      producer.send(message);

      connection.close();
    } catch (Exception ex) {
      System.out.println(ex.getMessage());
    }
```

}

d. Запустите клиентское приложение.

e. В консоли администрирования ActiveMQ на странице Queues убедитесь, что в нужную очередь было добавлено сообщение (столбец Number of Pending Messages). Для просмотра сообщения нажмите на ссылку с названием очереди, а затем на ссылку с идентификатором сообщения (пример отображаемой информации приведен ниже).

Headers ↑	
Correlation ID	
Destination	queue://replyQueue
Expiration	0
Group	
Message ID	ID:pulsar-59923-1322991829778-2:24:4:1:1
Persistence	Persistent
Priority	4
Redelivered	false
Reply To	
Sequence	0
Timestamp	2011-12-04 17:27:01:962 MSK
Type	

Message Actions

Delete

Copy

Move

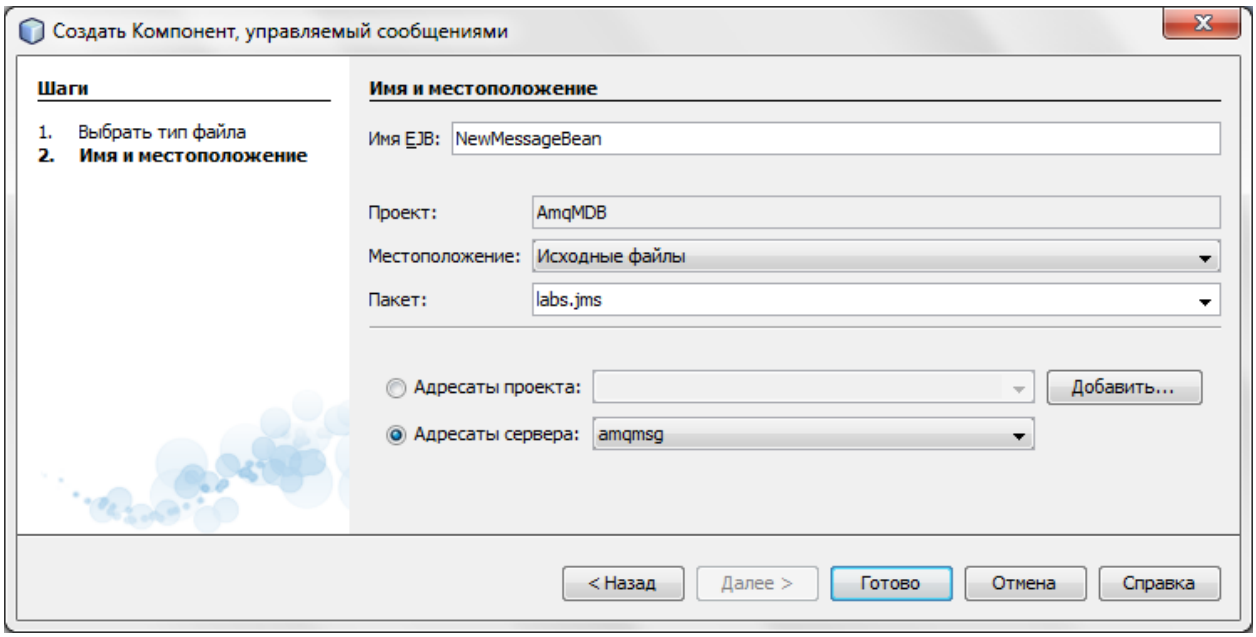
-- Please select --

Message Details

```
Test message
```

3) Создайте проект EJB-модуля с компонентом, управляемым сообщениями.

a. Создайте компонент, управляемый сообщениями. Выберите в списке адресатов сервера созданную ранее очередь.



b. Определите в классе компонента обработчик сообщений – метод onMessage().
   i. Для отправки ответного сообщения клиенту потребуется фабрика соединений с системой обмена сообщениями. Проще всего создать ее следующим образом: в контекстном меню редактора класса компонента выберите пункт «Вставка кода / Отправить сообщение JMS». В диалоге выберите в списке адресатов сервера созданную ранее очередь, а в поле «Фабрика подключений» укажите название ресурса коннектора. После того, как среда разработки сгенерирует методы для отправки сообщения, нужно удалить из класса поле типа Queue и заменить его на очередь, указанную клиентом в заголовке сообщения JMSReplyTo.

c. Для взаимодействия с ActiveMQ требуется дополнительная конфигурация компонента, управляемого сообщениями, которая выполняется в стандартном и специфическом дескрипторах. Создайте дескриптор для Glassfish (файл типа «GlassFish / Дескриптор GlassFish») и отредактируйте его исходный код на вкладке XML:

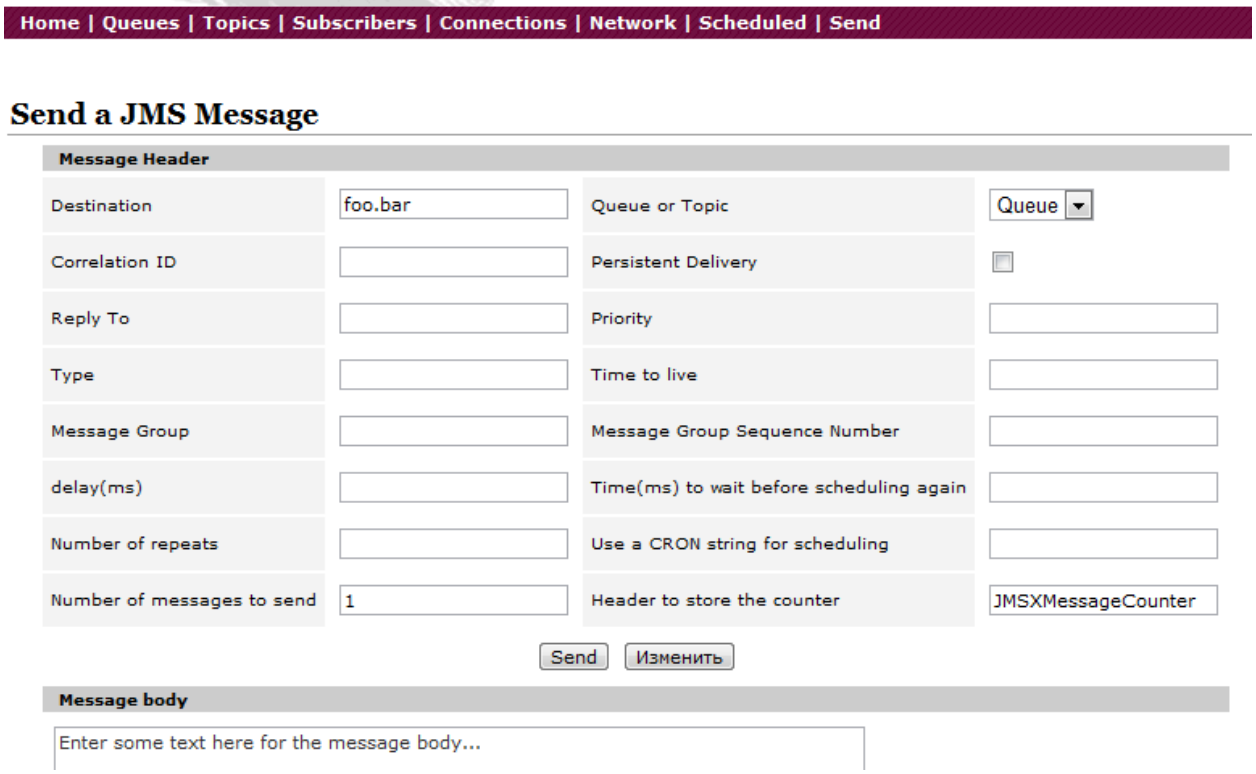
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE glassfish-ejb-jar PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 EJB 3.1//EN"
"http://glassfish.org/dtds/glassfish-ejb-jar_3_1-1.dtd">
<glassfish-ejb-jar>
 <enterprise-beans>
  <ejb>
   <ejb-name>название_MDB-компонента</ejb-name>
   <mdb-connection-factory>
    <jndi-name>название _ресурса_коннектора</jndi-name>
   </mdb-connection-factory>
   <mdb-resource-adapter>
    <resource-adapter-mid>activemq-rar-5.5.1</resource-adapter-mid>
   </mdb-resource-adapter>
  </ejb>
 </enterprise-beans>
</glassfish-ejb-jar>
```

d. Создайте стандартный дескриптор (файл типа «Enterprise JavaBeans / Стандартный дескриптор развертывания») и отредактируйте его:

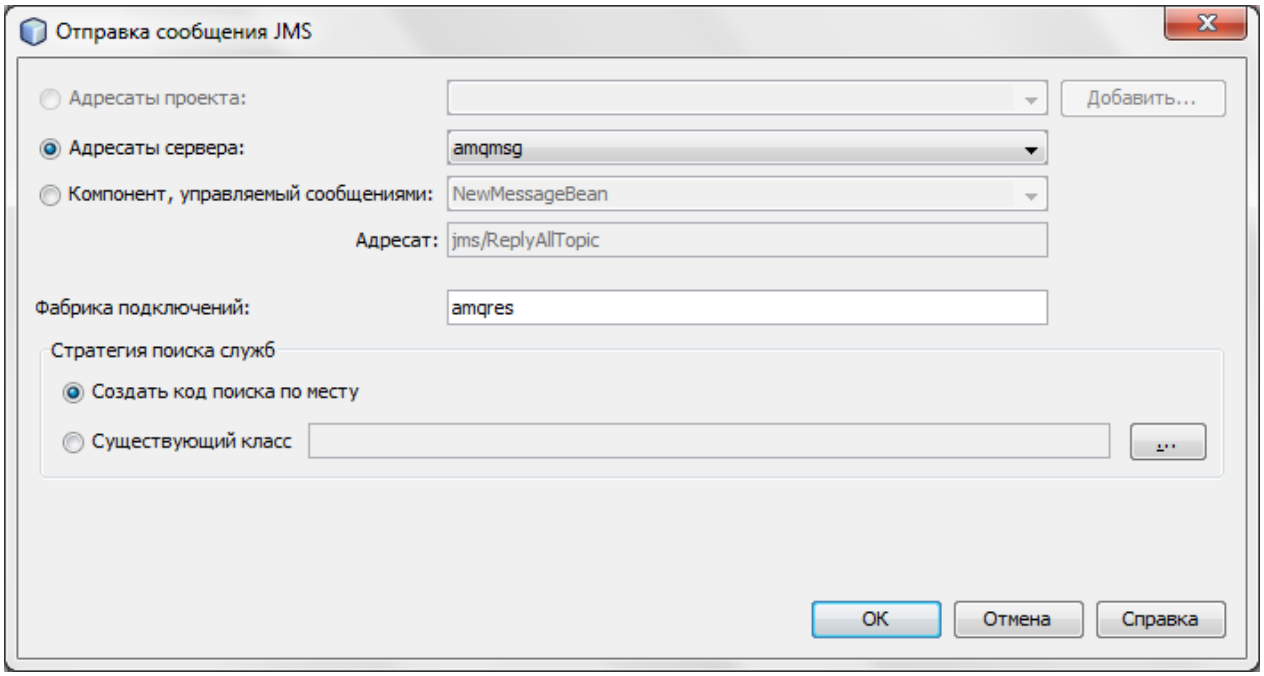
```
<?xml version="1.0" encoding="UTF-8"?>
<ejb-jar xmlns = "http://java.sun.com/xml/ns/javaee"
      version = "3.1"
      xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation = "http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/ejb-
jar_3_1.xsd" metadata-complete="false">
  <display-name>Ejb1</display-name>
  <enterprise-beans>
    <message-driven>
      <display-name>название_MDB-компонента</display-name>
      <ejb-name>название_MDB-компонента</ejb-name>
      <ejb-class>полное_имя_класса_MDB-компонента</ejb-class>
      <messaging-type>javax.jms.MessageListener</messaging-type>
      <transaction-type>Container</transaction-type>
      <activation-config>
        <activation-config-property>
          <activation-config-property-name>DestinationType</activation-config-property-name>
          <activation-config-property-value>javax.jms.Queue</activation-config-property-value>
        </activation-config-property>
        <activation-config-property>
          <activation-config-property-name>Destination</activation-config-property-name>
          <activation-config-property-value>название_очереди</activation-config-property-value>
        </activation-config-property>
      </activation-config>
    </message-driven>
  </enterprise-beans>
  <assembly-descriptor>
    <container-transaction>
      <method>
        <ejb-name>название_MDB-компонента</ejb-name>
        <method-name>onMessage</method-name>
        <method-params>
          <method-param>java.lang.String</method-param>
        </method-params>
      </method>
      <trans-attribute>Required</trans-attribute>
    </container-transaction>
  </assembly-descriptor>
</ejb-jar>
```

e. Установите проект EJB-модуля на сервере приложений. Убедитесь в консоли администрирования ActiveMQ, что компонент обработал сообщение, ранее отправленное из клиентского приложения (в столбце Number of Pending Messages должен быть 0).

f. Отправьте с помощью консоли администрирования ActiveMQ новое сообщение в очередь (пункт Send в верхнем меню, см. рис. ниже) и убедитесь, что компонент обработал это сообщение. Обратите внимание, что по мере добавления и обработки сообщений в таблице Queues консоли администрирования увеличивается значение счетчиков Messages Enqueued и Messages Dequeued, соответственно.



4) Создайте проект веб-приложения
   a. Создайте фэйслет (или JSP-страницу) для ввода данных и отправки сообщения.
   b. Создайте класс управляемого бина. В контекстном меню редактора исходного кода управляемого бина выберите пункт «Вставка кода / Отправить сообщение JMS». В диалоге выберите в списке адресатов сервера созданную ранее очередь, а в поле «Фабрика подключений» укажите название ресурса коннектора.



   c. Добавьте вызов сгенерированного средой метода отправки сообщения в метод обработки действия пользователя (action-м етод). Измените, при необходимости, сгенерированный средой метод для создания сообщения.

d. Запустите веб-приложение и убедитесь, что оно успешно отправляет сообщения в очередь.

5) Доработайте проект клиентского приложения (добавьте создание временной очереди и получение ответа от сервера в синхронном режиме).