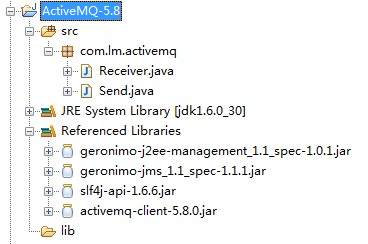
<http://heisetoufa.iteye.com/blog/1908335>

1.下载ActiveMQ   
去官方网站下载：<http://activemq.apache.org/>   
我下载的时候是 ActiveMQ 5.8.0 Release版   
  
2.运行ActiveMQ   
解压缩apache-activemq-5.8.0-bin.zip，然后双击apache-activemq-5.5.1\bin\activemq.bat运行ActiveMQ程序。   
  
启动ActiveMQ以后，登陆：http://localhost:8161/admin/，创建一个Queue，命名为FirstQueue。   
  
3.创建Eclipse项目并运行   
创建java project：ActiveMQ-5.8，新建lib文件夹   
打开apache-activemq-5.8.0\lib目录   
拷贝   
activemq-broker-5.8.0.jar   
activemq-client-5.8.0.jar   
geronimo-j2ee-management\_1.1\_spec-1.0.1.jar   
geronimo-jms\_1.1\_spec-1.1.1.jar   
slf4j-api-1.6.6.jar   
这5个jar文件到lib文件夹中,并Build Path->Add to Build Path   
  
结构如图:   
  
   
  
  
Sender.java

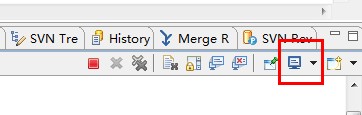
**Java代码  [收藏代码](javascript:void())**

1. **package** com.lm.activemq;
3. /\*\*
4. \* @Header: Sender.java
5. \* 类描述：
6. \* @author: lm
7. \* @date 2013-7-17 上午10:52:42
8. \* @Email
9. \* @company 欢
10. \* @addr 北京市朝阳区劲松
11. \*/
12. **import** javax.jms.Connection;
13. **import** javax.jms.ConnectionFactory;
14. **import** javax.jms.DeliveryMode;
15. **import** javax.jms.Destination;
16. **import** javax.jms.MessageProducer;
17. **import** javax.jms.Session;
18. **import** javax.jms.TextMessage;
19. **import** org.apache.activemq.ActiveMQConnection;
20. **import** org.apache.activemq.ActiveMQConnectionFactory;
22. **public** **class** Sender {
23. **private** **static** **final** **int** SEND\_NUMBER = 5;
25. **public** **static** **void** main(String[] args) {
26. // ConnectionFactory ：连接工厂，JMS 用它创建连接
27. ConnectionFactory connectionFactory; // Connection ：JMS 客户端到JMS
28. // Provider 的连接
29. Connection connection = **null**; // Session： 一个发送或接收消息的线程
30. Session session; // Destination ：消息的目的地;消息发送给谁.
31. Destination destination; // MessageProducer：消息发送者
32. MessageProducer producer; // TextMessage message;
33. // 构造ConnectionFactory实例对象，此处采用ActiveMq的实现jar
34. connectionFactory = **new** ActiveMQConnectionFactory(
35. ActiveMQConnection.DEFAULT\_USER,
36. ActiveMQConnection.DEFAULT\_PASSWORD, "tcp://localhost:61616");
37. **try** { // 构造从工厂得到连接对象
38. connection = connectionFactory.createConnection();
39. // 启动
40. connection.start();
41. // 获取操作连接
42. session = connection.createSession(Boolean.TRUE,
43. Session.AUTO\_ACKNOWLEDGE);
44. // 获取session注意参数值xingbo.xu-queue是一个服务器的queue，须在在ActiveMq的console配置
45. destination = session.createQueue("FirstQueue");
46. // 得到消息生成者【发送者】
47. producer = session.createProducer(destination);
48. // 设置不持久化，此处学习，实际根据项目决定
49. producer.setDeliveryMode(DeliveryMode.NON\_PERSISTENT);
50. // 构造消息，此处写死，项目就是参数，或者方法获取
51. sendMessage(session, producer);
52. session.commit();
53. } **catch** (Exception e) {
54. e.printStackTrace();
55. } **finally** {
56. **try** {
57. **if** (**null** != connection)
58. connection.close();
59. } **catch** (Throwable ignore) {
60. }
61. }
62. }
64. **public** **static** **void** sendMessage(Session session, MessageProducer producer)
65. **throws** Exception {
66. **for** (**int** i = 1; i <= SEND\_NUMBER; i++) {
67. TextMessage message = session.createTextMessage("ActiveMq 发送的消息"
68. + i);
69. // 发送消息到目的地方
71. System.out.println("发送消息：" + "ActiveMq 发送的消息" + i);
72. producer.send(message);
73. }
74. }
75. }

Receiver.java 

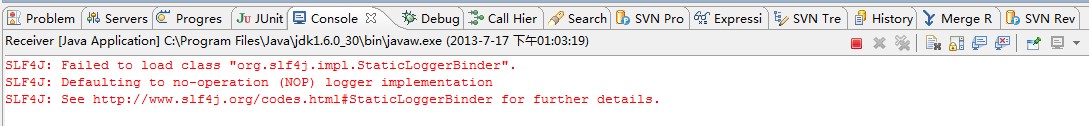
**Java代码  [收藏代码](javascript:void())**

1. **package** com.lm.activemq;
3. /\*\*
4. \* @Header: Receiver.java
5. \* 类描述：
6. \* @author: lm
7. \* @date 2013-7-17 上午10:52:58
8. \* @Email
9. \* @company 欢
10. \* @addr 北京市朝阳区劲松
11. \*/
12. **import** javax.jms.Connection;
13. **import** javax.jms.ConnectionFactory;
14. **import** javax.jms.Destination;
15. **import** javax.jms.MessageConsumer;
16. **import** javax.jms.Session;
17. **import** javax.jms.TextMessage;
18. **import** org.apache.activemq.ActiveMQConnection;
19. **import** org.apache.activemq.ActiveMQConnectionFactory;
21. **public** **class** Receiver {
22. **public** **static** **void** main(String[] args) {
23. // ConnectionFactory ：连接工厂，JMS 用它创建连接
24. ConnectionFactory connectionFactory;
25. // Connection ：JMS 客户端到JMS Provider 的连接
26. Connection connection = **null**;
27. // Session： 一个发送或接收消息的线程
28. Session session;
29. // Destination ：消息的目的地;消息发送给谁.
30. Destination destination;
31. // 消费者，消息接收者
32. MessageConsumer consumer;
33. connectionFactory = **new** ActiveMQConnectionFactory(
34. ActiveMQConnection.DEFAULT\_USER,
35. ActiveMQConnection.DEFAULT\_PASSWORD, "tcp://localhost:61616");
36. **try** {
37. // 构造从工厂得到连接对象
38. connection = connectionFactory.createConnection();
39. // 启动
40. connection.start();
41. // 获取操作连接
42. session = connection.createSession(Boolean.FALSE,
43. Session.AUTO\_ACKNOWLEDGE);
44. // 获取session注意参数值xingbo.xu-queue是一个服务器的queue，须在在ActiveMq的console配置
45. destination = session.createQueue("FirstQueue");
46. consumer = session.createConsumer(destination);
47. **while** (**true**) {
48. // 设置接收者接收消息的时间，为了便于测试，这里谁定为100s
49. TextMessage message = (TextMessage) consumer.receive(100000);
50. **if** (**null** != message) {
51. System.out.println("收到消息" + message.getText());
52. } **else** {
53. **break**;
54. }
55. }
56. } **catch** (Exception e) {
57. e.printStackTrace();
58. } **finally** {
59. **try** {
60. **if** (**null** != connection)
61. connection.close();
62. } **catch** (Throwable ignore) {
63. }
64. }
65. }
66. }

5.测试过程   
先运行:Receiver.java   
再运行:Sender.java   
  
可以看到结果   
Sender运行后:   
发送消息：ActiveMq 发送的消息1   
发送消息：ActiveMq 发送的消息2   
发送消息：ActiveMq 发送的消息3   
发送消息：ActiveMq 发送的消息4   
发送消息：ActiveMq 发送的消息5   
  
Receiver运行后:   
收到消息ActiveMq 发送的消息1   
收到消息ActiveMq 发送的消息2   
收到消息ActiveMq 发送的消息3   
收到消息ActiveMq 发送的消息4   
收到消息ActiveMq 发送的消息5   
  
要想看到不同的输出内容,通过点击如下图的按钮切换console   
  
   
  
  
在Receiver.java中，可以设置一个时间，比如receive(500000)，如下代码所示：

**Java代码  [收藏代码](javascript:void())**

1. TextMessage message = (TextMessage) consumer.receive(500000);

这个时候运行Receiver.java的话，会使得这个Receiver.java一直运行500秒，在eclipse中可以发现：   
  
  
  
  
点击那个红色方块可以手动停止运行程序   
  
黑色头发:http://heisetoufa.iteye.com/