# 简介

[JMS](http://www.baike.com/wiki/JMS)（[Java Message Service](http://www.baike.com/wiki/Java+Message+Service)，[Java消息服务](http://www.baike.com/wiki/Java%E6%B6%88%E6%81%AF%E6%9C%8D%E5%8A%A1)）是一组Java应用[程序](http://www.baike.com/wiki/%E7%A8%8B%E5%BA%8F)接口(Java [API](http://www.baike.com/wiki/API))，它提供创建、发送、接收、读取[消息](http://www.baike.com/wiki/%E6%B6%88%E6%81%AF)的服务。由Sun公司和它的合作伙伴设计的JMS API定义了一组公共的应用程序接口和相应语法，使得Java程序能够和其他消息[组件](http://www.baike.com/wiki/%E7%BB%84%E4%BB%B6)进行通信。

  JMS是一种与厂商无关的 API，用来访问消息收发系统。它[类](http://www.baike.com/wiki/%E7%B1%BB)似于 [JDBC](http://www.baike.com/wiki/JDBC) (Java [Database](http://www.baike.com/wiki/Database) Connectivity)：这里，JDBC 是可以用来访问许多不同关系[数据库](http://www.baike.com/wiki/%E6%95%B0%E6%8D%AE%E5%BA%93)的 API，而 JMS 则提供同样与厂商无关的访问方法，以访问消息收发服务。许多厂商目前都支持 JMS，包括 IBM 的 MQSeries、BEA的 Weblogic JMS service和 Progress 的 [SonicMQ](http://www.baike.com/wiki/SonicMQ)，这只是几个例子。

JMS 使您能够通过消息收发服务（有时称为消息中介程序或路由器）从一个 JMS 客户机向另一个 JML 客户机发送消息。消息是 JMS 中的一种[类型](http://www.baike.com/wiki/%E7%B1%BB%E5%9E%8B)[对象](http://www.baike.com/wiki/%E5%AF%B9%E8%B1%A1)，由两部分组成：报头和消息主体。报头由路由信息以及有关该消息的[元数据](http://www.baike.com/wiki/%E5%85%83%E6%95%B0%E6%8D%AE)组成。消息主体则携带着应用程序的数据或有效负载。根据有效负载的类型来划分，可以将消息分为几种类型，它们分别携带：简单文本 (TextMessage)、可序列化的对象 (ObjectMessage)、属性集合 (MapMessage)、字节流 (BytesMessage)、原始值流 (StreamMessage)，还有无有效负载的消息 (Message)。

# 模型：

|  |  |  |
| --- | --- | --- |
| 发送者 | MQServer | 接收者 |

P2P，一对一模型

发布订阅， 一对多模型

# 数据持久化：

<http://blog.csdn.net/xyw_blog/article/details/9128219>

* 可以使用远程数据库，配置jdbc

一是直接的JDBC操作，二是journal的jdbc操作。Journal是优化过的，先存文件，然后再批量写数据库。

* 文件存储KahaDB， 具体查看activemq.xml

# 协议：

发送者和接收者都长连接到MQServer。

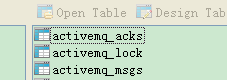
支持的数据类型有：JMS 定义了5中消息类型： TextMessage、MapMessage、BytesMessage、

StreamMessage和ObjectMessage

选择最合适的消息类型可以使JMS最有效 的处理消息。

# 配置：

配置mysql持久化

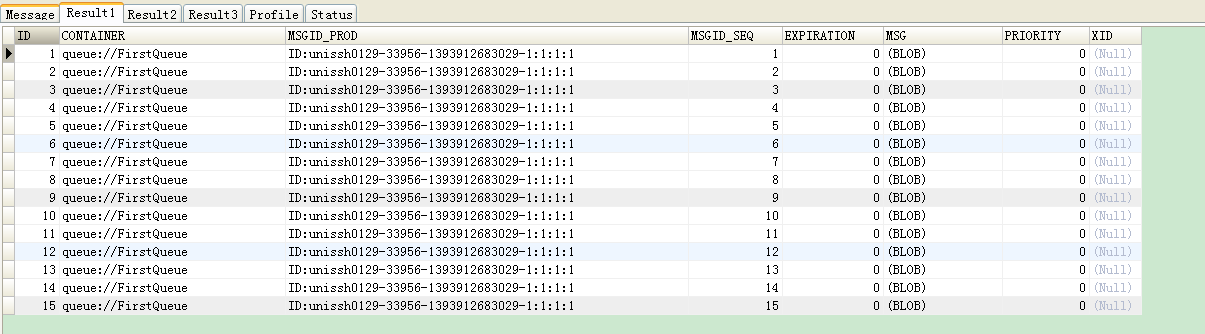


配置成功后查看数据库会发现多了3张表。

具体配置查看

**activemq.xml 和 jdbc.properties**

# 测试：



* producer.setDeliveryMode(DeliveryMode.PERSISTENT); // 消息设置持久化

发送15条数据后发现这15条数据已经作为离线消息存入数据库了。

* 当开始一个receiver后，成功接收到了消息，然后再查数据库就发现数据已经删了。

# 远程管理：

<!--

The managementContext is used to configure how ActiveMQ is exposed in

JMX. By default, ActiveMQ uses the MBean server that is started by

the JVM. For more information, see:

参考 http://activemq.apache.org/jmx.html

-->

<broker useJmx="true" brokerName="BROKER1">

</broker>

# activemqbrowser

在http://sourceforge.net/projects/activemqbrowser/上下载了个 activemqbrowser，远程管理工具。按照<http://activemq.apache.org/jmx.html> 配置后，死活连接不上，而用java自带的jConsole却可以连上去的。 叫我不得怀疑activemqbrowser这工具是不是有问题。

经过不懈的折腾发现：run\_activemq\_browser.bat 使用的5.6的jar。

|  |
| --- |
| **echo off**  **start "ActiveMQBrowser" javaw -Xms128m -Xmx512m -splash:cube.png -cp .\QBrowserV2\_Neo.jar;.\activemq-web-5.6.0.jar;.\activemq-all-5.6.0.jar;.\jide-oss-2.6.2.jar;.\imq.jar;.\jms.jar;.\imqadmin\_ja.jar;.\imqadmin.jar;.\imqutil\_ja.jar;.\imqutil.jar;.\imqjmx.jar;.\imqjmx\_ja.jar com.qbrowser.ActiveMQBrowser** |

而我安装的Activemq版本是5.9.0， 尝试将activemq-all-5.6.0.jar和activemq-web-5.6.0.jar替换为activemq-all-5.9.0.jar， activemq-web-5.9.0.jar。OK，居然OK了。

这是版本问题，搞定！！！！！

最后提醒一点，如果启动activemq过程中，出现

Error: Password file read access must be restricted: D:\apache-activemq-5.0.0\bin\../conf/  
jmx.password

参考 <http://eric-gcm.iteye.com/blog/1399500>， 修改文件访问属性即可解决！