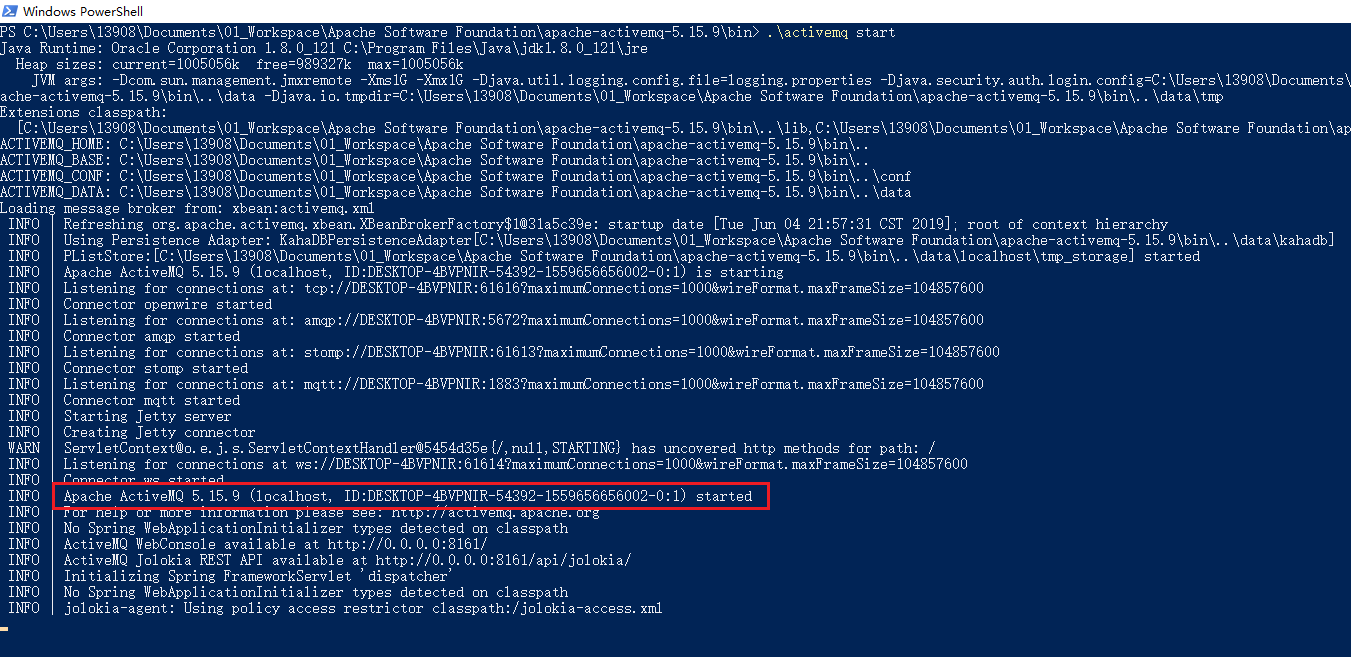
# Windows下ActiveMQ安装与配置

1. 下载zip包
2. 将压缩包放到指定目录下，解压。
3. 启动：打开cmd，进入到%ACTIVEMQ\_HOME%\bin目录下，执行命令：

activemq start //打开服务

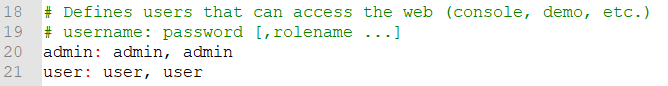
看到started表示启动成功。

1. 测试：浏览器进入到127.0.0.1:8161，activemq控制台，可查看activemq中的各消息信息；

用户名：admin

密码：admin

用户名和密码%ACTIVEMQ\_HOME%\conf中的jetty-realm.properties中配置。



1. 启动中可能会遇到的问题及解决方案

Q1：端口被占用

A1：activemq的默认端口为：61616

#修改activemq的默认端口：

进入到%ACTIVEMQ\_HOME%\conf中修改activemq.xml配置文件。

Q2：将%ACTIVEMQ\_HOME%\bin写入环境变量中

A2：可避免每次启动服务需cmd进入指定文件夹下命令操作。

# Linux下ActiveMQ安装与配置

1. 下载Activemq，并解压到指定的目录。

|  |
| --- |
| tar xzvf apache-activemq-5.15.9-bin.tar.gz |

1. 启动Activemq

|  |
| --- |
| ./activemq start  netstat -anp|grep 61616 |

1. 打开web管理页面

http://127.0.0.1:8161/admin

admin/admin

# ActiveMQ的基本概念

## JMS消息模型

|  |
| --- |
| 点对点模型（Queue） |
| 发布者/订阅者模型（Topic） |

* 点对点模型（Queue）

一个生产者向一个特定的队列发布消息，一个消费者从这个队列中依次读取消息。

模型特点：**只有一个消费者获得消息**。

* 发布者/订阅者模型（Topic）

0个或多个订阅者可以接受特定主题的消息。

模型特点：**多个消费者可以获得消息**。

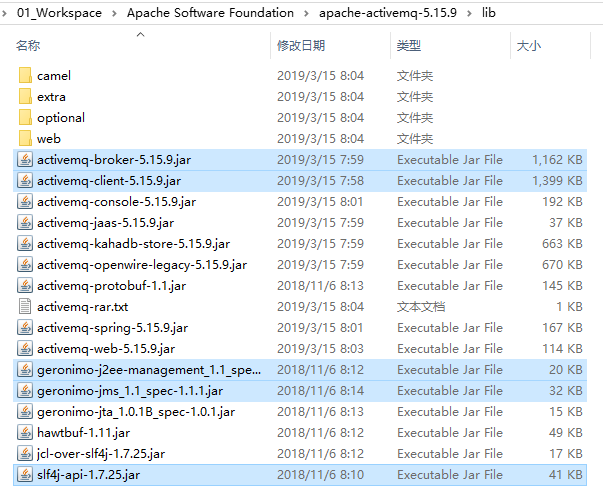
|  |
| --- |
| Topic是以广播的形式，通知所有在线监听的客户端有新的消息，没有监听的客户端将收不到消息。 |
| Queue是以点对点的形式通知多个处于监听状态的客户端中的一个。 |

## JMS消息格式

|  |  |
| --- | --- |
| MapMessage | key-value键值对 |
| TextMessage | 字符串对象 |
| ObjectMessage | 一个序列化的Java对象 |
| ByteMessage | 一个未解释字节的数据流 |
| StreamMessage | Java原始值的数据流 |

# 编写Java使用MQ的Demo实例

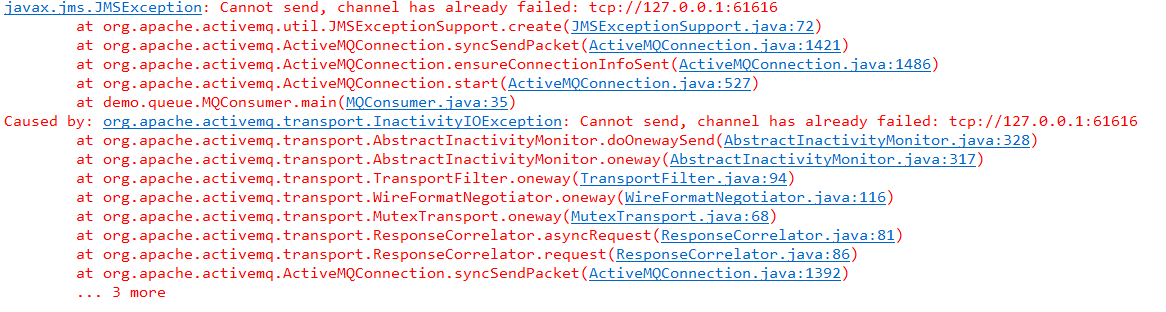
1. 需要引入Activemq需要的jar包，从lib路径下找。

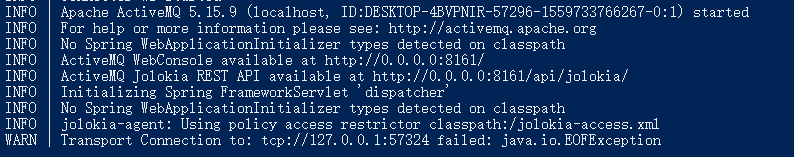


1. 创建Session时的参数说明：

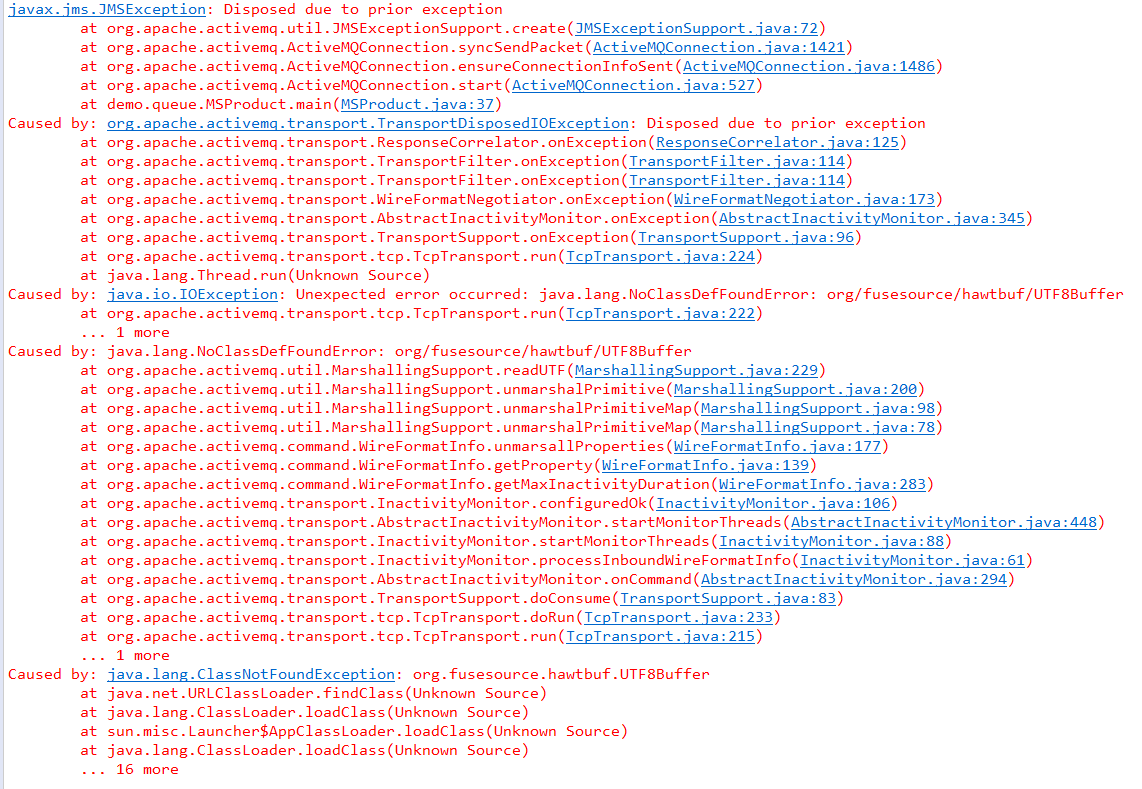
|  |  |
| --- | --- |
| Session.***AUTO\_ACKNOWLEDGE*** |  |
| Session.***CLIENT\_ACKNOWLEDGE*** |  |
| Session.***DUPS\_OK\_ACKNOWLEDGE*** |  |
| Session.***SESSION\_TRANSACTED*** |  |

1. 运行消费者程序的时候出现的异常：





1. 运行生产者Demo程序的时候出现异常



引入jar包“hawtbuf-1.11.jar”即可解决上述的问题。

1. 点对点通信需要注意的地方：

可以启动多个消费者进行监听，但是只有其中一个可以正常进行消费。

1. 生产者在发送消息的过程中，只要Connection的连接没有被close掉，生产者可以一直发送消息。
2. Activemq开发过程中需要引入的jar包

|  |  |  |
| --- | --- | --- |
| groupId | artifactId | version |
| org.apache.activemq | activemq-broker | 5.15.9 |
| org.apache.activemq | activemq-client | 5.15.9 |
| org.apache.geronimo.specs | geronimo-j2ee-management\_1.1\_spec | 1.0.1 |
| org.apache.geronimo.specs | geronimo-jms\_1.1\_spec | 1.1.1 |
| org.fusesource.hawtbuf | hawtbuf | 1.11 |
| org.slf4j | slf4j-api | 1.7.25 |

# Spring与ActiveMQ的集成

# JMS基础知识点

JMS（Java消息服务），两个应用程序之间进行异步通信的API，为标准消息协议和消息服务提供了一组通用接口，包括创建，发送，读取消息等，用于支持JAVA应用程序开发。在J2EE中，当两个应用程序使用JMS进行通信时，它们之间并不是直接相连的，而是通过一个共同的消息收发服务连接起来，可以达到解耦的效果。

Apache ActiveMQ™ is the most popular open source, multi-protocol, Java-based messaging server. It supports industry standard protocols so users get the benefits of client choices across a broad range of languages and platforms. Connectivity from C, C++, Python, .Net, and more is available. Integrate your multi-platform applications using the ubiquitous **AMQP** protocol. Exchange messages between your web applications using **STOMP** over websockets. Manage your IoT devices using **MQTT**. Support your existing **JMS** infrastructure and beyond. ActiveMQ offers the power and flexibility to support any messaging use-case.

Apache ActiveMQ是最受欢迎的，开源的，支持多种协议，基于Java的消息服务器。它支持工业标准协议，因此可以给用户在选择大范围的跨语言，跨平台的客户端时提供便利。通过C，C++，Python，.Net以及更多的语言进行连接。使用无处不在的AMQP协议集成你的多平台应用程序。使用基于websocket的STOMP协议在你的web应用程序之间交换信息。使用MQTT协议管理你的IoTa设备。支持你的已经存在的JMS基础设施。ActiveMQ提供最有力的，最灵活的，对任何消息使用案例的支持。

## ActiveMQ、RabbitMQ、RocketMQ和Kafka比较

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 特性 | ActiveMQ | RabbitMQ | RocketMQ | Kafka |
| 开发语言 | Java | erlang | Java | scala |
| 单机吞吐量 | 万级 | 万级 | 10万级 | 10万级 |
| 时效性 | ms级 | ms级 | ms级 | ms级以内 |
| 可用性 | 高（主从架构） | 高（主从架构） | 非常高（分布式架构） | 非常高（分布式架构） |
| 功能特性 | 成熟的产品，在很多公司得到应用；有较多的文档；各种协议支持较好 | 基于erlang开发，所以并发能力很强，性能极其好，延时很低；管理界面较丰富 | MQ功能比较完备，扩展性佳 | 只支持主要的MQ功能，像一些消息查询，消息回溯等功能没有提供，毕竟是为大数据准备的，在大数据领域应用广。 |

## 消息查询

## 消息回溯