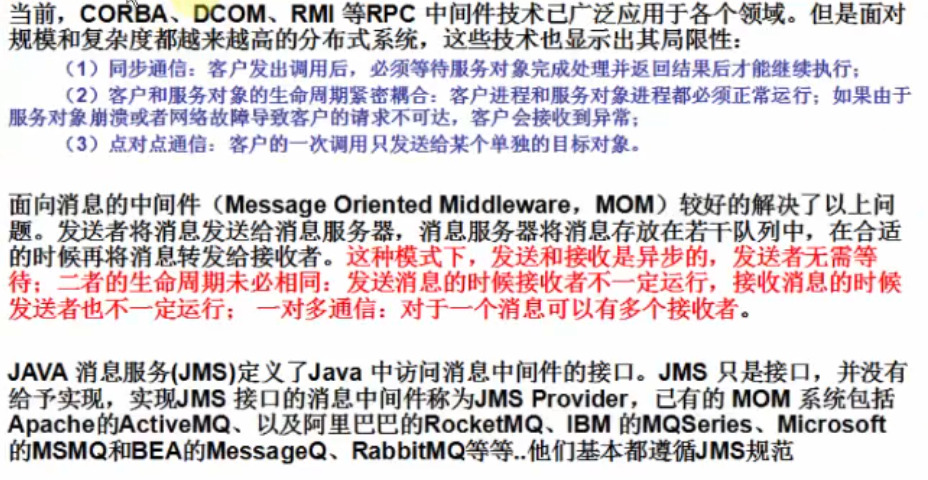
ActiveMQ

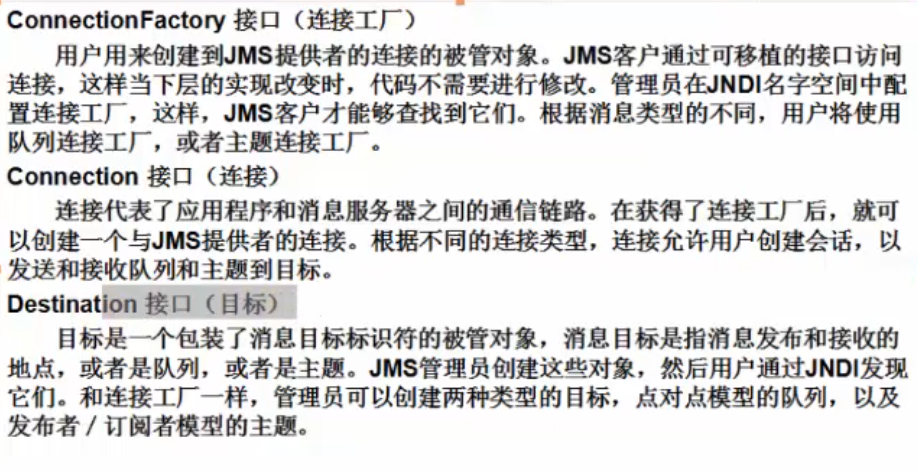
# ActiveMQ简介和入门

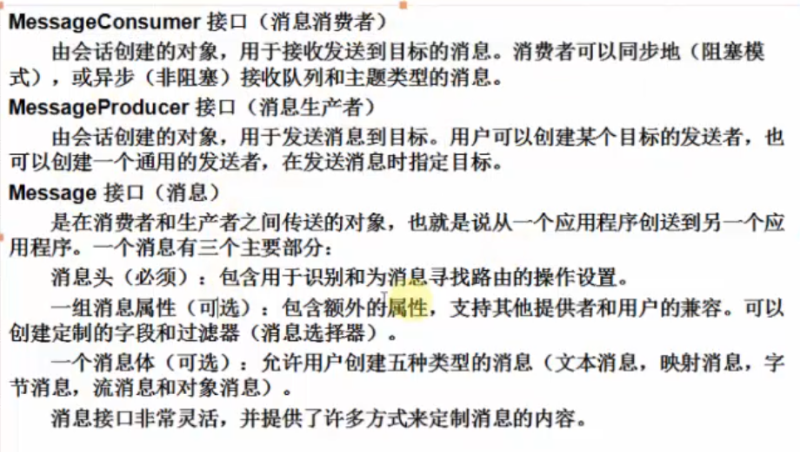
## 背景&JMS概述

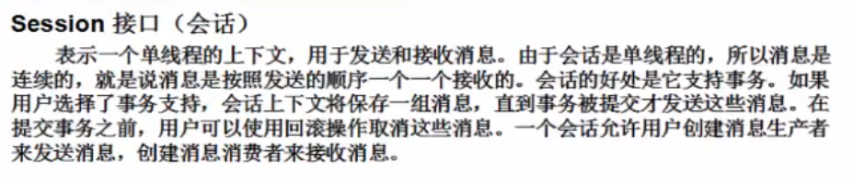


## JMS术语

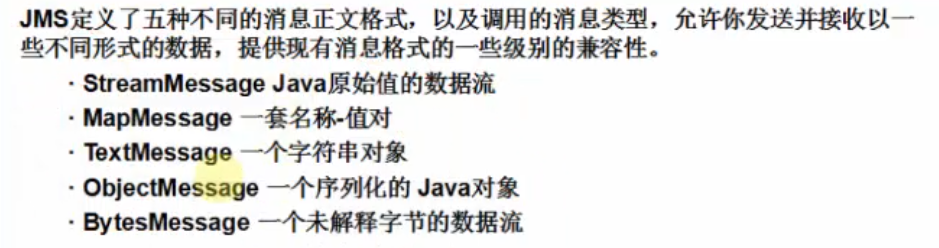




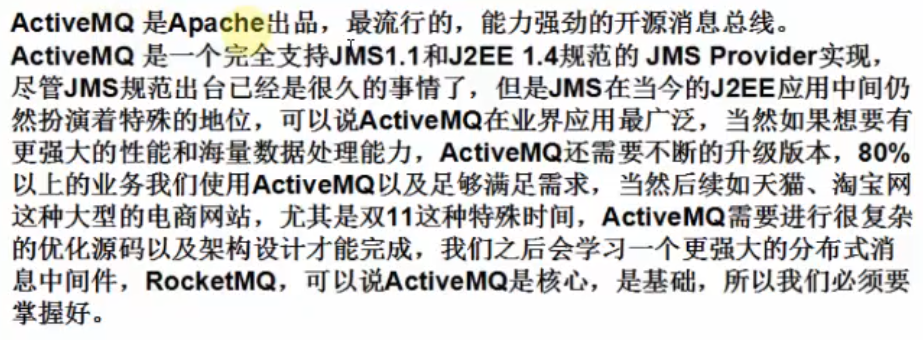




## JMS消息格式



## ActiveMQ简介



## ActiveMQ的安装

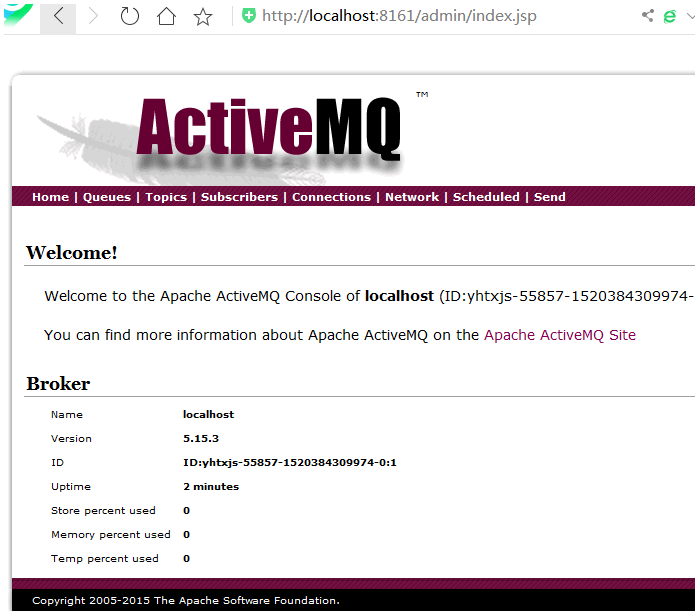
去官网下载<http://activemq.apache.org/>

我们下载最新的版本5.15.3

先使用window版本

ActiveMQ解压后动后的目录





# ActiveMQ的使用

## ActiveMQ-HelloWorld



### JMSProducer

**package** com.guigu.activemq.helloworld;

**import** javax.jms.Connection;

**import** javax.jms.ConnectionFactory;

**import** javax.jms.DeliveryMode;

**import** javax.jms.Destination;

**import** javax.jms.JMSException;

**import** javax.jms.MessageProducer;

**import** javax.jms.Session;

**import** javax.jms.TextMessage;

**import** org.apache.activemq.ActiveMQConnectionFactory;

/\*\*

\* <p>project\_name:ActiveMQProvider</p>

\* <p>package\_name:com.guigu.activemq.helloworld.JMSProducer</p>

\* <p>description：</p>

\* <p>@author：刘老师<p>

\* <p> date:2018年3月7日上午9:19:59 </p>

\* <p>comments： </p>

\* <p>@version jdk1.8</p>

\*

\* <p>Copyright (c) 2018, 980991634@qq.com All Rights Reserved. </p>

\*/

**public** **class** JMSProducer {

**public** **static** **void** main(String[] args) {

**try** {

// 1.建立ConnectionFactory工厂对象 ，需要填入用户名 密码 以及连接的地址 都使用默认即可

// 默认端口号是tcp://localhost:61616

ConnectionFactory connectionFactory = **new** ActiveMQConnectionFactory(ActiveMQConnectionFactory.***DEFAULT\_USER***,

ActiveMQConnectionFactory.***DEFAULT\_PASSWORD***, "tcp://localhost:61616");

// 2.通过ConnetionFactory工厂 创建一个Connection连接

// 并且调用Connection的start方法开启连接

Connection connection = connectionFactory.createConnection();

connection.start();

// 3.通过Connection对象创建Session会话 上下文件环境对象 用于接受消息 参数1代表 是否是开启事务 参数2签收方式

// 一般使用自动签收

Session session = connection.createSession(Boolean.***FALSE***, Session.***AUTO\_ACKNOWLEDGE***);

// 4.通过sesssion创建Destination对象 指的是一个客户端用来指定生产消息目标和消费消息来源的对象

Destination destination = session.createQueue("HelloWolrd\_guigu");

// 5.通过Session对象 创建消息的发送和接受对象(生产者和消费者)

MessageProducer producer = session.createProducer(destination);

// 6.我们可以使用MessageProducer对象的setDeliveryMode方法设置持久化特性 和非持久化特性

producer.setDeliveryMode(DeliveryMode.***NON\_PERSISTENT***);

// 发送消息到activemq消息服务器

**for** (**int** i = 0; i < 10; i++) {

TextMessage msg = session.createTextMessage("我是消息内容:" + i + "activemq is so easy");

producer.send(msg);

System.***out***.println("发送消息：" + msg.getText());

Thread.*sleep*(1000);

}

// 7.释放连接即可

**if** (connection != **null**) {

connection.close();

}

} **catch** (JMSException e) {

e.printStackTrace();

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

}

### JMSConsumer

**package** com.guigu.activemq.helloworld;

**import** javax.jms.Connection;

**import** javax.jms.ConnectionFactory;

**import** javax.jms.DeliveryMode;

**import** javax.jms.Destination;

**import** javax.jms.JMSException;

**import** javax.jms.MessageConsumer;

**import** javax.jms.MessageProducer;

**import** javax.jms.Session;

**import** javax.jms.TextMessage;

**import** org.apache.activemq.ActiveMQConnectionFactory;

/\*\*

\* <p>project\_name:ActiveMQProvider</p>

\* <p>package\_name:com.guigu.activemq.helloworld.JMSConsumer</p>

\* <p>description：</p>

\* <p>@author：刘老师<p>

\* <p> date:2018年3月7日上午10:16:43 </p>

\* <p>comments： </p>

\* <p>@version jdk1.8</p>

\*

\* <p>Copyright (c) 2018, 980991634@qq.com All Rights Reserved. </p>

\*/

**public** **class** JMSConsumer {

**public** **static** **void** main(String[] args) {

**try** {

// 1.建立ConnectionFactory工厂对象 ，需要填入用户名 密码 以及连接的地址 都使用默认即可

// 默认端口号是tcp://localhost:61616

ConnectionFactory connectionFactory = **new** ActiveMQConnectionFactory(ActiveMQConnectionFactory.***DEFAULT\_USER***,

ActiveMQConnectionFactory.***DEFAULT\_PASSWORD***, "tcp://localhost:61616");

// 2.通过ConnetionFactory工厂 创建一个Connection连接

// 并且调用Connection的start方法开启连接

Connection connection = connectionFactory.createConnection();

connection.start();

// 3.通过Connection对象创建Session会话 上下文件环境对象 用于接受消息 参数1代表 是否是开启事务 参数2签收方式

// 一般使用自动签收

Session session = connection.createSession(Boolean.***FALSE***, Session.***AUTO\_ACKNOWLEDGE***);

// 4.通过sesssion创建Destination对象 指的是一个客户端用来指定生产消息目标和消费消息来源的对象

Destination destination = session.createQueue("HelloWolrd\_guigu");

// 5.通过Session对象 创建消息的发送和接受对象(生产者和消费者)

MessageConsumer consumer = session.createConsumer(destination);

**while**(**true**) {

TextMessage msg =(TextMessage) consumer.receive();

**if**(msg==**null**) {

**break**;

}

System.***out***.println("收到内容:"+msg.getText());

}

// 7.释放连接即可

**if** (connection != **null**) {

connection.close();

}

} **catch** (JMSException e) {

e.printStackTrace();

}

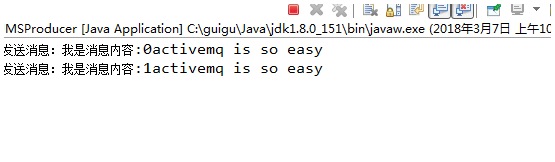
}

}

验证

1. 首先启动ActiveMQ 在浏览器上<http://localhost:8161/amdin>

2、运行生产者



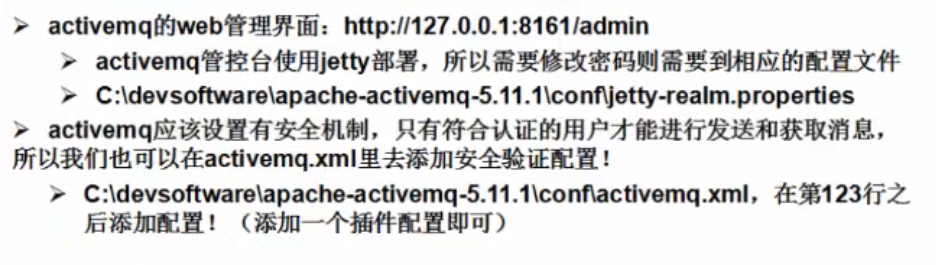
查看ActiveMQ服务器相关的信息

1. 运行消费者

查看ActiveMQ服务器相关的信息s



## ActiveMQ安全机制

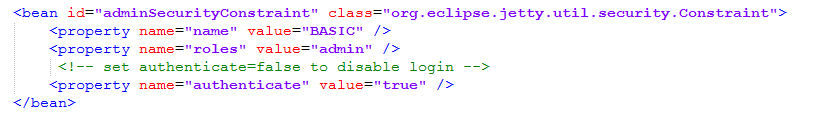


**一、设置后台管理密码**

**A）修改jetty.xml开启密码验证**

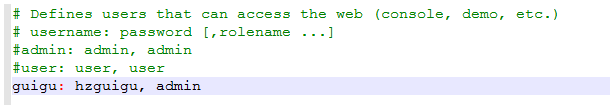
**C:\guigu\apache-activemq-5.15.3-bin\apache-activemq-5.15.3\conf\jetty.xml文件**

在低版本中默认是false是没有开启的 。



**B) 修改控制台登录用户名和密码 也可也不修改 默认就是admin admin**

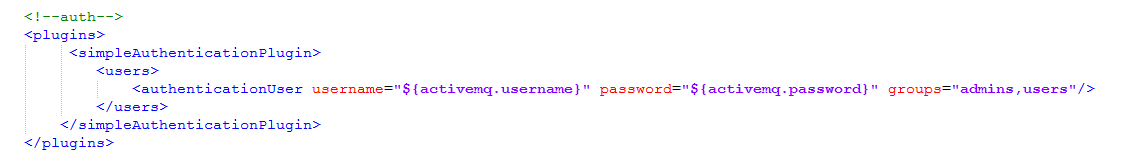
jetty-realm.properties



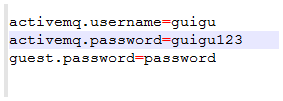
**二、消息生产者和消息消费者密码认证**

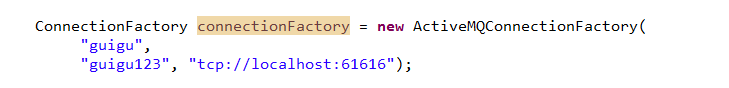
**A)添加权限认证**

C:\guigu\apache-activemq-5.15.3-bin\apache-activemq-5.15.3\conf\activemq.xml

 **B)设置用户名和密码**

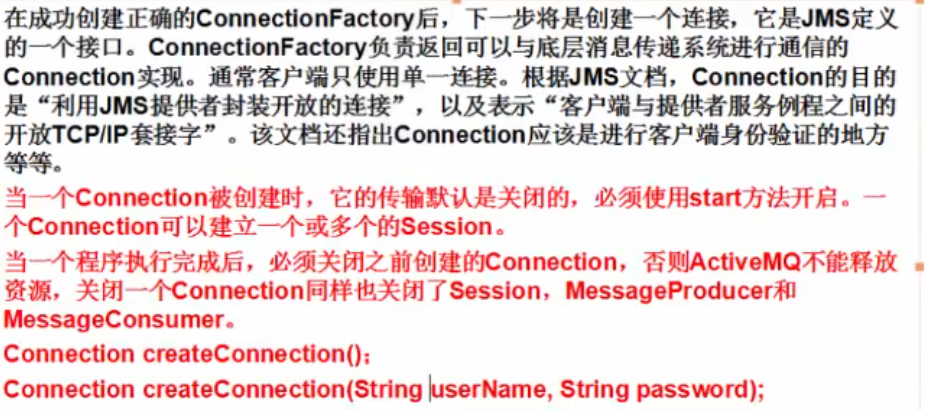
credentials.properties





## API详解

### Connection方法的使用



### Session方法的使用



**JMSProducer**

// 3.通过Connection对象创建Session会话 上下文件环境对象 用于接受消息 参数1代表 是否是开启事务 参数2签收方式

// 一般使用自动签收

// Session session = connection.createSession(Boolean.FALSE, Session.AUTO\_ACKNOWLEDGE);

//开启事务 需要手动提交 签收模式为 手动签收

Session session = connection.createSession(Boolean.***TRUE***, Session.***CLIENT\_ACKNOWLEDGE***);

// 4.通过sesssion创建Destination对象 指的是一个客户端用来指定生产消息目标和消费消息来源的对象

Destination destination = session.createQueue("HelloWolrd\_guigu");

// 5.通过Session对象 创建消息的发送和接受对象(生产者和消费者)

MessageProducer producer = session.createProducer(destination);

// 6.我们可以使用MessageProducer对象的setDeliveryMode方法设置持久化特性 和非持久化特性

producer.setDeliveryMode(DeliveryMode.***NON\_PERSISTENT***);

// 发送消息到activemq消息服务器

**for** (**int** i = 0; i < 10; i++) {

TextMessage msg = session.createTextMessage("我是消息内容:" + i + "activemq is so easy");

producer.send(msg);

System.***out***.println("发送消息：" + msg.getText());

Thread.*sleep*(1000);

}

//事务开启 必须手动提交事务

session.commit();

**JMSConsumer**

// 4.通过sesssion创建Destination对象 指的是一个客户端用来指定生产消息目标和消费消息来源的对象

Destination destination = session.createQueue("HelloWolrd\_guigu");

// 5.通过Session对象 创建消息的发送和接受对象(生产者和消费者)

MessageConsumer consumer = session.createConsumer(destination);

**while**(**true**) {

TextMessage msg =(TextMessage) consumer.receive();

**if**(msg==**null**) {

**break**;

}

//手动确认接收消息

**msg.acknowledge();**

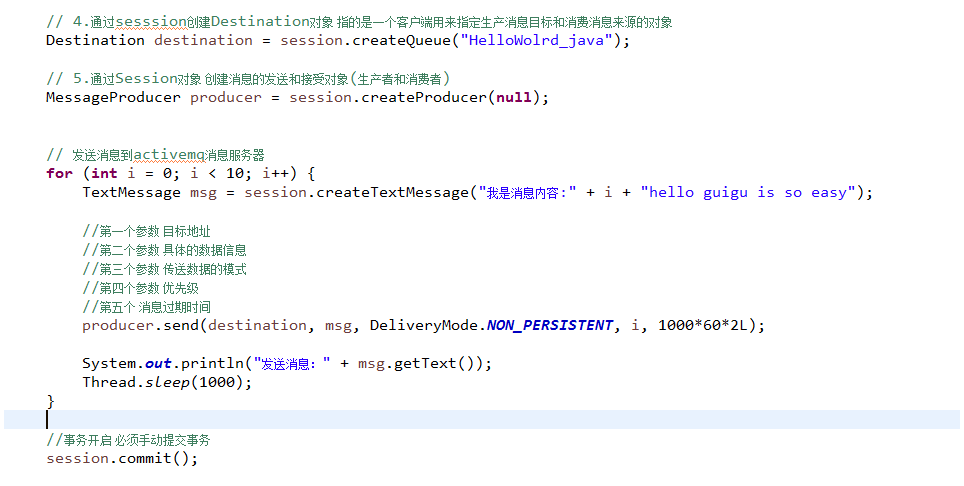
System.***out***.println("收到内容:"+msg.getText());

}

### MessageProducer



**JMSProducer2**



**JMSConsumer2**



### MessageConsumer（配置消息选择器）



**package** com.guigu.activemq.messageselector;

/\*\*

\* <p>project\_name:ActiveMQProvider</p>

\* <p>package\_name:com.guigu.activemq.messageselector.Producer</p>

\* <p>description：</p>

\* <p>@author：刘老师<p>

\* <p> date:2018年3月7日下午1:57:41 </p>

\* <p>comments： </p>

\* <p>@version jdk1.8</p>

\*

\* <p>Copyright (c) 2018, 980991634@qq.com All Rights Reserved. </p>

\*/

**import** javax.jms.Connection;

**import** javax.jms.ConnectionFactory;

**import** javax.jms.DeliveryMode;

**import** javax.jms.Destination;

**import** javax.jms.JMSException;

**import** javax.jms.MapMessage;

**import** javax.jms.MessageProducer;

**import** javax.jms.Session;

**import** org.apache.activemq.ActiveMQConnectionFactory;

**public** **class** Producer {

// 1.连接工厂

**private** ConnectionFactory connectionFactory;

// 2.连接对象

**private** Connection connection;

// 3.Sesssion对象

**private** Session session;

// 4.定义生产者

**private** MessageProducer messageProducer;

**public** Producer() {

**try** {

**this**.connectionFactory = **new** ActiveMQConnectionFactory("tcp://localhost:61616");

**this**.connection = connectionFactory.createConnection("guigu", "guigu123");

**this**.connection.start();

// 设置自动签收模式

**this**.session = connection.createSession(**false**, Session.***AUTO\_ACKNOWLEDGE***);

**this**.messageProducer = session.createProducer(**null**);

} **catch** (JMSException e) {

e.printStackTrace();

}

}

**public** Session getSession() {

**return** **this**.session;

}

**public** **void** send1() {

**try** {

Destination destination =**this**.session.createQueue("first");

MapMessage msg1 =session.createMapMessage();

msg1.setString("name", "guiguhz");

msg1.setInt("age", 30);

//设置用于消息过滤的条件

msg1.setStringProperty("name", "gghz");

msg1.setIntProperty("age", 30);

msg1.setStringProperty("color", "blue");

MapMessage msg2 =session.createMapMessage();

msg2.setString("name", "zhangsan");

msg2.setInt("age", 25);

//设置用于消息过滤的条件

msg2.setStringProperty("name", "zs");

msg2.setIntProperty("age", 25);

msg2.setStringProperty("color", "white");

MapMessage msg3 =session.createMapMessage();

msg3.setString("name", "guiguhz");

msg3.setInt("age", 35);

//设置用于消息过滤的条件

msg3.setStringProperty("name", "zl");

msg3.setIntProperty("age", 35);

msg3.setStringProperty("color", "black");

//发送消息到消息中间件

**this**.messageProducer.send(destination,msg1,DeliveryMode.***NON\_PERSISTENT***,4,1000\*60\*10);

**this**.messageProducer.send(destination,msg2,DeliveryMode.***NON\_PERSISTENT***,4,1000\*60\*10);

**this**.messageProducer.send(destination,msg3,DeliveryMode.***NON\_PERSISTENT***,4,1000\*60\*10);

} **catch** (JMSException e) {

e.printStackTrace();

}

}

**public** **static** **void** main(String[] args) {

Producer p1 =**new** Producer();

p1.send1();

}

}

**package** com.guigu.activemq.messageselector;

**import** javax.jms.Connection;

**import** javax.jms.ConnectionFactory;

**import** javax.jms.Destination;

**import** javax.jms.JMSException;

**import** javax.jms.MapMessage;

**import** javax.jms.Message;

**import** javax.jms.MessageConsumer;

**import** javax.jms.MessageListener;

**import** javax.jms.Session;

**import** javax.jms.TextMessage;

**import** org.apache.activemq.ActiveMQConnectionFactory;

/\*\*

\* <p>project\_name:ActiveMQProvider</p>

\* <p>package\_name:com.guigu.activemq.messageselector.Consumer</p>

\* <p>description：</p>

\* <p>@author：刘老师<p>

\* <p> date:2018年3月7日下午2:14:03 </p>

\* <p>comments： </p>

\* <p>@version jdk1.8</p>

\*

\* <p>Copyright (c) 2018, 980991634@qq.com All Rights Reserved. </p>

\*/

**public** **class** Consumer {

// 1.连接工厂

**private** ConnectionFactory connectionFactory;

// 2.连接对象

**private** Connection connection;

// 3.Sesssion对象

**private** Session session;

// 4.定义消费者对象

**private** MessageConsumer messageConsumer;

// 5.目的地

**private** Destination destination;

// 定义消息选择器规则

**public** **final** String SELECTOR\_1 = "age > 28 and color='black'";

**public** Consumer() {

**try** {

**this**.connectionFactory = **new** ActiveMQConnectionFactory("tcp://localhost:61616");

**this**.connection = connectionFactory.createConnection("guigu", "guigu123");

**this**.connection.start();

// 设置自动签收模式

**this**.session = connection.createSession(**false**, Session.***AUTO\_ACKNOWLEDGE***);

**this**.destination = session.createQueue("first");

**this**.messageConsumer = session.createConsumer(destination, SELECTOR\_1);

} **catch** (JMSException e) {

e.printStackTrace();

}

}

**public** Session getSession() {

**return** **this**.session;

}

// 用于监听消息队列的消息

**class** MyListener **implements** MessageListener {

@Override

**public** **void** onMessage(Message message) {

System.***out***.println("-------33");

**try** {

// 首先判断数据类型

**if** (message **instanceof** MapMessage) {

System.***out***.println("-------11");

MapMessage result = (MapMessage) message;

System.***out***.println(result.toString());

System.***out***.println(result.getString("name"));

System.***out***.println(result.getInt("age"));

// System.out.println(result.getInt("color"));

}

} **catch** (JMSException e) {

e.printStackTrace();

}

}

}

//用于异步监听消息

**public** **void** receiver() {

**try** {

messageConsumer.setMessageListener(**new** MyListener());

System.***out***.println("-------22");

} **catch** (JMSException e) {

e.printStackTrace();

}

}

**public** **static** **void** main(String[] args) {

Consumer consumer =**new** Consumer();

consumer.receiver();

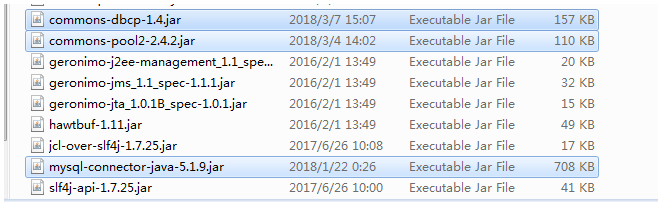
}

}

## ActiveMQ持久化

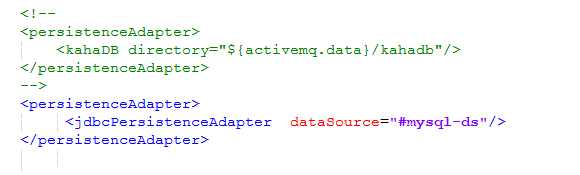
**一）把ActiveMQ 持久化到mysql**

1、把数据库的jar包 放入到activemq的lib目录下



1. 修改activemq的conf目录下的activemq.xml配置文件

2.1修改原来默认的持久化方式改为mysql 默认的kahadab注释



2.2连接mysql的配置

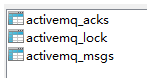


1. 重启activemq服务器即可
2. 配置持久化操作



当消息没有被消费会同步暂存到mysql数据库内。

1. 数据库的内容



activemq\_msgs用于存储消息，Queue和Topic都存储在这个表中：

ID：自增的数据库主键

CONTAINER：消息的Destination

MSGID\_PROD：消息发送者客户端的主键

MSG\_SEQ：是发送消息的顺序，MSGID\_PROD+MSG\_SEQ可以组成JMS的MessageID

EXPIRATION：消息的过期时间，存储的是从1970-01-01到现在的毫秒数

MSG：消息本体的Java序列化对象的二进制数据

PRIORITY：优先级，从0-9，数值越大优先级越高

activemq\_acks用于存储订阅关系。如果是持久化Topic，订阅者和服务器的订阅关系在这个表保存：

主要的数据库字段如下：

CONTAINER：消息的Destination

SUB\_DEST：如果是使用Static集群，这个字段会有集群其他系统的信息

CLIENT\_ID：每个订阅者都必须有一个唯一的客户端ID用以区分

SUB\_NAME：订阅者名称

SELECTOR：选择器，可以选择只消费满足条件的消息。条件可以用自定义属性实现，可支持多属性AND和OR操作

LAST\_ACKED\_ID：记录消费过的消息的ID。

表activemq\_lock在集群环境中才有用，只有一个Broker可以获得消息，称为Master Broker，

其他的只能作为备份等待Master Broker不可用，才可能成为下一个Master Broker。

这个表用于记录哪个Broker是当前的Master Broker。

## ActiveMQ-发布订阅消息模式

## ActiveMQ的具体业务使用

# ActiveMQ集群