**消息中间件使用与开发**

1. **消息中间件部署**
2. **安装与配置**

这里主要说windows环境，linux环境除了启动执行sh之外和windows是一样的

系统运行环境必须是jdk1.7 或者以上，建议1.7。

下载activemq-5.13.3

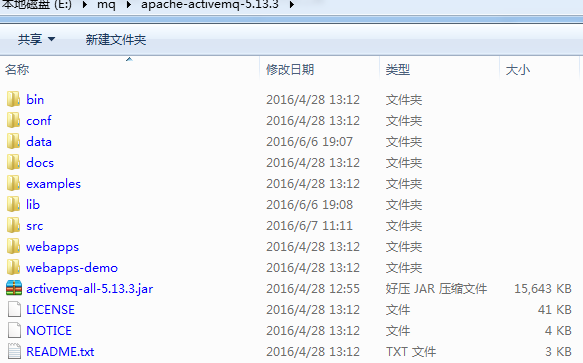
Windows：

ftp://192.168.5.10/软件工具/中间件/apache-activemq-5.13.3-bin.zip

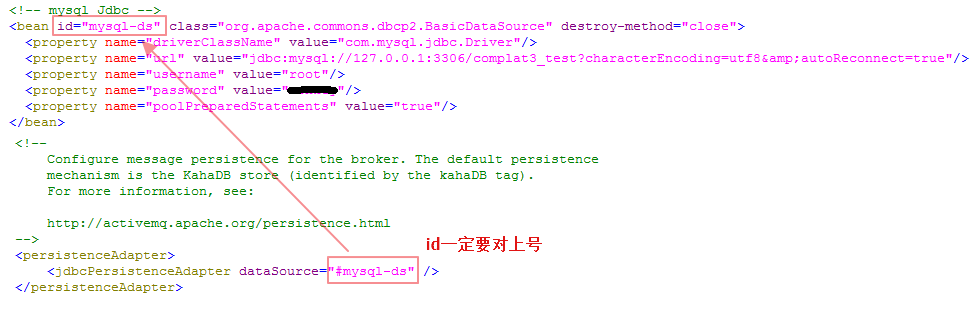
Linux：

ftp://192.168.5.10/软件工具/中间件/apache-activemq-5.13.3-bin.zip

解压出来放到非中文目录



修改配置conf/activemq.xml，让mq的消息可以使用数据库，具体配置可以参照examples/conf/activemq-jdbc.xml



需要将对应的数据库驱动jar包放到lib目录中

1. **运行**

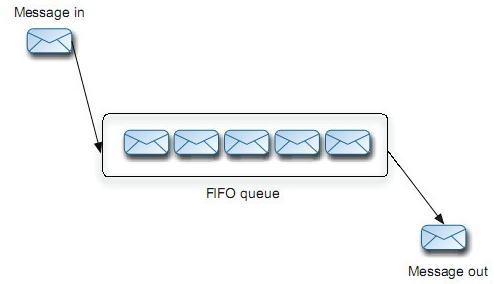
直接运行：bin/activemq.bat start

启动如果不报错访问 <http://127.0.0.1:8161/admin/> 如果可以看到控制台说明允许成功。

1. **平台3中使用消息**

**使用消息中间件activemq ，中间件运行环境一定要是jdk1.7的（开发环境可以是1.6）**

1. **队列**

****

一个或者多个发送方向一个管道中发送消息，一个或者多个接受方从管道中获取消息

管道中的消息顺序按照送达先后排列，接收方按照消息先后顺序一条一条的拿走，多个接收方按照各自的速度（手快）竞争拿到消息处理，已经被拿走的消息不会被其他接收方再次获得，管道中的消息拿走一条就少一条。

如果接受方不在线，管道中的消息会持久化保存，即使消息中间件重启也不会丢失。

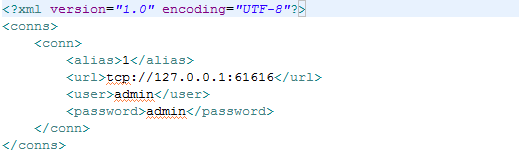
适合做多系统联合处理，比如多机发布，一个网站几十万的信息要发布，通过队列分10台机器一起发布，效率能明显提高。

**编码**

* **准备**

将平台中的目录optional/mq/WEB-INF/lib下所有jar包拷贝到工程的lib

配置WebContent/WEB-INF/config/mq.xml，这个xml是用来连接消息中间件的，当数据库连接配置来理解。



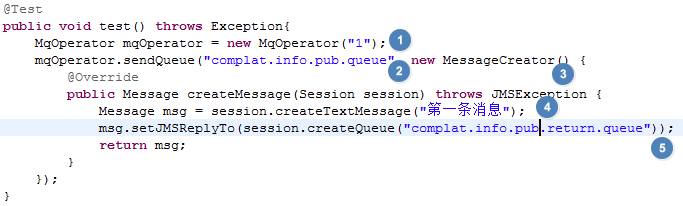
conns：所有消息中间件的连接

conn：一个消息中间件的连接

alias：连接别名（很重要，不能重复）

url、user、password：你懂的

* **发送端**



①使用MqOperator来操作发送，构造方法中的“1”表示连接别名。

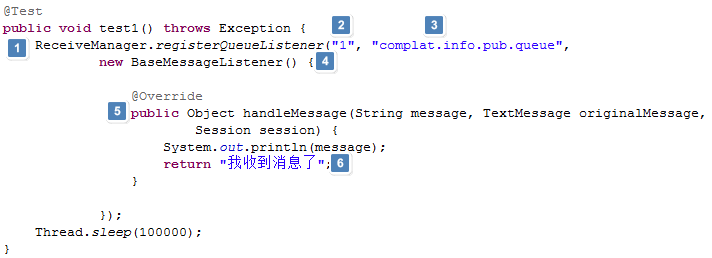
②sendQuery表示发送消息至队列，第一个参数是队列名称（目的地），名称的规则为“应用名称.模块名称.事项.类型（类型有2个，队列为queue，发布为topic）”

③一个接口主要用来构造消息。

④产生一个消息，通过session可以产生5种类型的消息，最常用的是文本消息

⑤（可选项）消息被使用后需要接受方反馈的目的地

* **接收端**



①使用ReceiveManager来接收消息，接收消息实际上是个监听只需要注册一次就可以了，接收队列用registerQueueListener。

②第一个参数的“1”表示连接别名

③第二个参数表示接收消息的来源（队列名称）

④一个基本类BaseMessageListener用来处理接收的消息，他有6个方法可以按照消息类型来重写

⑤处理消息的方法

⑥反馈消息，如果发送方指定的反馈的目的地，这里就按照目的地发消息，如果返回null，不会发出反馈消息（即使发送发给了反馈的目的地）

1. **发布/订阅**

与队列类似但又有很大区别。说个场景

消息发送方发送了条消息，现在需要3个接收方都要有这条消息，这个场景很常见，比如我们的3个系统数据需要同步信息，我们系统录了一条信息，需要给搜索系统，要给微门户，要给社会化分享平台。

这时候队列用一条少一条就做不到了。

想解决上述场景我们可能想到个现实的情景，就是广播，拿着大喇叭对着3人喊，这时候3人都被通知到了。那么问题来了如果有个人不在场就通知不到了，现在假设这个不在场的人在现场放了个录音机，就能解决这个问题了。这3个在场和不在场放了录音机的人实际上就在做订阅的行为，大喇叭在做发布的行为。

在简单点的比方就是报刊亭了，报刊亭每天都有好多报纸（发布），路人A每天都路过报刊亭拿起一份当天的看（临时订阅），但哪天没路过再次路过的时候也只能看当天的，没路过的那天的报纸就看不到了；路人B在报刊亭定了报纸（持久化订阅），报刊亭负责把每天的报纸都留了一份，你来了我就都给你。

**编码**

* **准备**

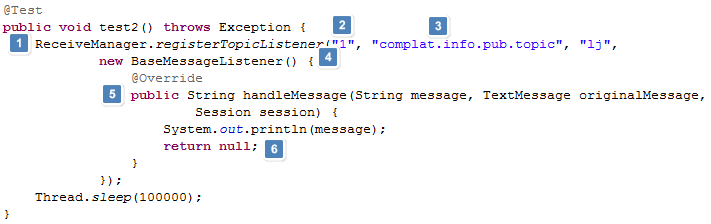
同队列

* **发布主题**



发布主题与队列的发布一样，只有方法名称不同，队列是sendQueue，这里是sendTopic，注意主题的目的地命名规则。

* **订阅主题**



注意：这种方式来订阅主题的消息属于临时订阅，也就是说不能收到历史的。

①使用ReceiveManager来接收消息，接收消息实际上是个监听只需要注册一次就可以了，接收队列用registerTopicListener。

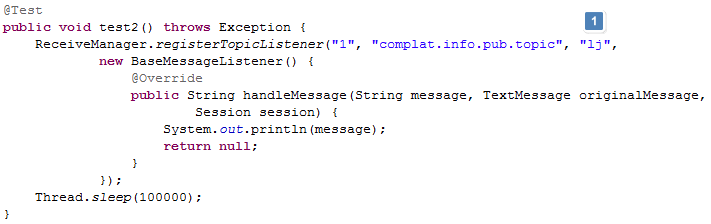
②第一个参数的“1”表示连接别名

③第二个参数表示接收消息的来源（主题名称）

④一个基本类BaseMessageListener用来处理接收的消息，他有6个方法可以按照消息类型来重写。

⑤处理消息的方法

⑥反馈消息，如果发送方指定的反馈的目的地，这里就按照目的地发消息，如果返回null，不会发出反馈消息（即使发送发给了反馈的目的地）



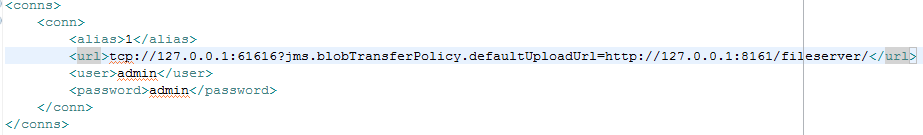
注意看这里，这个参数指定了用户名，就是告诉消息中间件，我这个用户的订阅是需要持久化的，即使我不在线下次我来的时候把历史消息要给我。

1. **附件消息**
2. **配置**

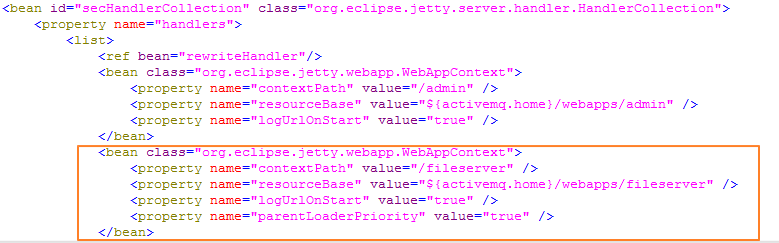
消息中间件对附件的处理是通过文件服务器来进行的。

消息与附件是分开发送的，流程是：上传附件到文件服务器->发送消息->接收消息->下载附件->消息处理

首先要配置文件服务器地址，在WebContent/WEB-INF/config/mq.xml中的url加上jms.blobTransferPolicy.defaultUploadUrl=http://127.0.0.1:8161/fileserver/



接下来需要配置文件服务器fileserver，在activeMq中已经集成了文件服务器但并没有开启（实际上配置是被注释掉了，放开即可），现在我们需要将它开启，修改activeMq目录下的conf/jetty.xml。



由于fileserver的类和web.xml有bug所以还需要把修改过的class和web.xml丢到fileserver里面，把平台的Document/培训资料/消息中间件/ custom /中的文件对应的直接覆盖到activeMq目录下的webapps\fileserver\中，到此就配置好了。

* **发送附件消息（以队列为例）**



* **接收附件消息（以队列为例）**



this.removeFile(originalMessage);

1. **其他说明**
2. **发送消息类型与接收消息重写方法对照的对照**

|  |  |
| --- | --- |
| Message | BaseMessageListener重写的方法 |
| BytesMessage | handleMessage(BytesMessage,x,x) |
| MapMessage | handleMessage(Map,x,x) |
| ObjectMessage | handleMessage(Serializable,x,x) |
| BlobMessage | handleMessage(InputStream,x,x) |
| TextMessage | handleMessage(String x,x,x) |
| Message | handleMessage(Message,x,x) |

1. **接收消息选择器**

接收方可以对所接收的消息进行选取，前提是发送方发送消息时对消息的属性进行了设置。



①设置了发送的消息type属性为1

②设置了发送的消息columnName属性为“领导信息”

③接受消息的时候可以在最后一个参数设置消息选择器，类似于sql

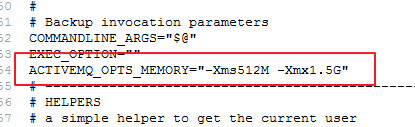
1. **遇到的问题**

出现大量Channel was inactive for too (>30000) long

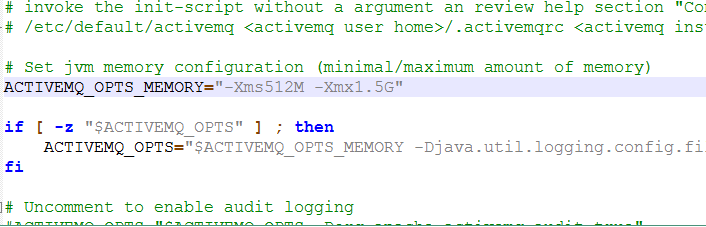
wireFormat.maxInactivityDuration=0

**加大mq内存（有两种情况）**

Linux 下在/mq目录/bin/activemq文件中加入ACTIVEMQ\_OPTS\_MEMORY="-Xms512M -Xmx1.5G"



如果没有生效，看看/mq目录/bin/是否有env文件，有的话，在这个文件中加入ACTIVEMQ\_OPTS\_MEMORY="-Xms512M -Xmx1.5G"



**集群配置**

failover:(tcp://primary:61616,tcp://secondary:61616)?xxx=xxx

规则为 failover(url,url,url)?参数

**传输大消息不行的解决**

Mysql有自己的包大小限制

show VARIABLES like '%max\_allowed\_packet%';

默认为1M

**修改方式1**

可以编辑my.cnf来修改（windows下my.ini）,在[mysqld]段或者[mysql](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "_blank)的server配置段进行修改。

max\_allowed\_packet = 20M

如果找不到my.cnf可以通过

mysql --help | grep my.cnf

**修改方式2**

set global max\_allowed\_packet = 2\*1024\*1024\*10

然后关闭掉这此mysql server链接，再进入。

show VARIABLES like '%max\_allowed\_packet%';

查看下max\_allowed\_packet是否编辑成功