# ActiveMQ分享6--JMS消息类型

# 1.前言

       之前的例子中，发送和接受的消息类型都是TextMessage，即文本消息（如下面的代码所示）。显然消息类型只有文本类型是不能满足要求的。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | //发送文本消息    session.createTextMessage(msg);      //接受文本消息    public  void  onMessage(Message msg) {       TextMessage message = (TextMessage) msg;         ……    } |

   根据Message接口的方法，可以获取消息类型

|  |  |
| --- | --- |
| 1 | String msgType = getJMSType() |

    根据 [JSR 914: JavaTM Message Service (JMS) API](https://jcp.org/aboutJava/communityprocess/final/jsr914/index.html) ，JMS规范中的消息类型包括TextMessage、BytesMessage、MapMessage、StreamMessage和ObjectMessage 等五种。ActiveMQ也有对应的实现，下面我们结合Spring JMS分别来看一下五种消息类型的收发代码。

# 2. 消息发送示例代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67 | /\*\*      \* 向默认队列发送text消息      \*/     public  void  sendMessage( final  String msg) {       String destination = jmsTemplate.getDefaultDestination().toString();       System.out.println( "ProducerService向队列"  + destination +  "发送了消息：\t"  + msg);       jmsTemplate.send( new  MessageCreator() {         public  Message createMessage(Session session)  throws  JMSException {           return  session.createTextMessage(msg);         }       });     }       /\*\*      \* 向默认队列发送map消息      \*/     public  void  sendMapMessage() {       jmsTemplate.send( new  MessageCreator() {         public  Message createMessage(Session session)  throws  JMSException {           MapMessage message = session.createMapMessage();           message.setString( "name" ,  "小西山" );           return  message;         }       });     }       /\*\*      \* 向默认队列发送Object消息      \*/     public  void  sendObjectMessage() {       jmsTemplate.send( new  MessageCreator() {         public  Message createMessage(Session session)  throws  JMSException {           Staff staff =  new  Staff( 1 ,  "搬砖工" );  // Staff必须实现序列化           ObjectMessage message = session.createObjectMessage(staff);           return  message;         }       });     }       /\*\*      \* 向默认队列发送Bytes消息      \*/     public  void  sendBytesMessage() {       jmsTemplate.send( new  MessageCreator() {         public  Message createMessage(Session session)  throws  JMSException {           String str =  "BytesMessage 字节消息" ;           BytesMessage message = session.createBytesMessage();           message.writeBytes(str.getBytes());           return  message;         }       });     }       /\*\*      \* 向默认队列发送Stream消息      \*/     public  void  sendStreamMessage() {       jmsTemplate.send( new  MessageCreator() {         public  Message createMessage(Session session)  throws  JMSException {           String str =  "StreamMessage 流消息" ;           StreamMessage message = session.createStreamMessage();           message.writeString(str);           message.writeInt( 521 );           return  message;         }       });     } |

# 3.消息接受示例代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43 | /\*\*      \* 接受消息      \*/     public  void  receive(Destination destination)  throws  JMSException {       Message message = jmsTemplate.receive(destination);       // 如果是文本消息       if  (message  instanceof  TextMessage) {         TextMessage tm = (TextMessage) message;         System.out.println( "ConsumerService从队列"  + destination.toString() +  "收到了消息：\t"  + tm.getText());       }         // 如果是Map消息       if  (message  instanceof  MapMessage) {         MapMessage mm = (MapMessage) message;         System.out.println( "ConsumerService从队列"  + destination.toString() +  "收到了消息：\t"             + mm.getString( "name" ));       }         // 如果是Object消息       if  (message  instanceof  ObjectMessage) {         ObjectMessage om = (ObjectMessage) message;         Staff staff = (Staff) om.getObject();         System.out.println( "ConsumerService从队列"  + destination.toString() +  "收到了消息：\t"  + staff);       }         // 如果是bytes消息       if  (message  instanceof  BytesMessage) {         byte [] b =  new  byte [ 1024 ];         int  len = - 1 ;         BytesMessage bm = (BytesMessage) message;         while  ((len = bm.readBytes(b)) != - 1 ) {           System.out.println( new  String(b,  0 , len));         }       }         // 如果是Stream消息       if  (message  instanceof  StreamMessage) {         StreamMessage sm = (StreamMessage) message;         System.out.println(sm.readString());         System.out.println(sm.readInt());       }       } |