## spring 整合 activemq 发送 MQ 消息[Topic]实例

1、pom 引入 <dependency> <groupId>junit <artifactId>junit</artifactId> <version>4.11 <scope>test</scope> </dependency> <dependency> <groupId>org.apache.activemq</groupId> <artifactId>activemg-all</artifactId> <version>5.11.1 </dependency> <dependency> <groupId>org.springframework <artifactId>spring-jms</artifactId> <version>4.1.4.RELEASE </dependency> <dependency> <groupId>org.springframework <artifactId>spring-test</artifactId> <version>4.1.4.RELEASE </dependency> 2、生产者配置 <?xml version="1.0" encoding="UTF-8"?> <beans xmlns="http://www.springframework.org/schema/beans"</pre> xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd"> <!-- 配置 JMS 连接工厂 --> <bean id="providerConnectionFactory"</pre> class="org.apache.activemq.ActiveMQConnectionFactory"> cproperty name="brokerURL" value="failover:(tcp://192.168.147.131:61616)" /> cproperty name="useAsyncSend" value="true" /> cproperty name="clientID" value="providerClienctConnect" /> </bean> <!-- 定义消息 Destination --> <bean id="topicDestination" class="org.apache.activemq.command.ActiveMQTopic"> <constructor-arg value="testSpringTopic"/> </bean> <!-- 消息发送者客户端 --> <bean id="providerJmsTemplate" class="org.springframework.jms.core.JmsTemplate"> property name="connectionFactory" ref="providerConnectionFactory" /> cproperty name="defaultDestination" ref="topicDestination" /> <!-- 开启订阅模式 --> property name="pubSubDomain" value="true"/> cproperty name="receiveTimeout" value="10000" /> <!-- deliveryMode, priority, timeToLive 的开关要生效,必须配置为 true,默认 false--> cproperty name="explicitQosEnabled" value="true"/> <!-- 发送模式 DeliveryMode.NON PERSISTENT=1:非持久; DeliveryMode.PERSISTENT=2:持久

```
cproperty name="deliveryMode" value="1"/>
    </bean>
</beans>
生产者程序
package com.mq.spring.topic;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.jms.core.JmsTemplate;
import org.springframework.jms.core.MessageCreator;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
import javax.annotation.Resource;
import javax.jms.*;
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations={"classpath:spring-topic.xml"})
public class TopicSender {
    @Resource(name = "providerJmsTemplate")
    private JmsTemplate jmsTemplate;
    @Test
    public void send() {
        sendMqMessage(null,"spring activemg topic type message[with listener] !");
     * 说明:发送的时候如果这里没有显示的指定 destination.将用 spring xml 中配置的 destination
     * @param destination
     * @param message
     */
    public void sendMqMessage(Destination destination, final String message) {
        if(null == destination) {
            destination = jmsTemplate.getDefaultDestination();
        jmsTemplate.send(destination, new MessageCreator() {
            public Message createMessage(Session session) throws JMSException {
                return session.createTextMessage(message);
        });
        System.out.println("spring send text message...");
    }
    public void setJmsTemplate(JmsTemplate jmsTemplate) {
        this.jmsTemplate = jmsTemplate;
3、消费者配置
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"</pre>
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd">
```

```
<!-- 配置 JMS 连接工厂 -->
   <bean id="consumerConnectionFactory"</pre>
class="org.apache.activemq.ActiveMQConnectionFactory">
       cproperty name="brokerURL" value="failover:(tcp://192.168.147.131:61616)" />
       cproperty name="useAsyncSend" value="true" />
       cproperty name="clientID" value="consumerClienctConnect" />
   </bean>
   <!-- 定义消息 Destination -->
   <bean id="topic1Destination" class="org.apache.activemq.command.ActiveMQTopic">
       <constructor-arg value="testSpringTopic1"/>
   </bean>
   <!-- 配置消息消费监听者 -->
   <bean id="consumerMessageListener"</pre>
class="com.mq.spring.topic.ConsumerMessageListener" />
   <!-- 消息订阅客户端 1 -->
   <bean id="consumerListenerClient1"</pre>
class="org.springframework.jms.listener.DefaultMessageListenerContainer">
       cproperty name="connectionFactory" ref="consumerConnectionFactory" />
       <!-- 开启订阅模式 -->
       cproperty name="pubSubDomain" value="true"/>
       cproperty name="destination" ref="topic1Destination" />
       cproperty name="subscriptionDurable" value="true"/>
       <!---这里是设置接收客户端的 ID, 在持久化时, 但这个客户端不在线时, 消息就存在数据库里, 直到被这个
ID 的客户端消费掉-->
       cproperty name="clientId" value="consumerClient1"/>
       <!-- 消息应答方式
           Session.AUTO ACKNOWLEDGE 消息自动签收
           Session.CLIENT ACKNOWLEDGE 客户端调用 acknowledge 方法手动签收
           Session.DUPS OK ACKNOWLEDGE 不必必须签收,消息可能会重复发送
       cproperty name="sessionAcknowledgeMode" value="1"/>
   </bean>
   <!-- 消息订阅客户端 2 -->
   <bean id="consumerListenerClient2"</pre>
class="org.springframework.jms.listener.DefaultMessageListenerContainer">
       cproperty name="connectionFactory" ref="consumerConnectionFactory" />
       <!-- 开启订阅模式 -->
       cproperty name="pubSubDomain" value="true"/>
       cproperty name="subscriptionDurable" value="true"/>
       <!---这里是设置接收客户端的 ID,在持久化时,但这个客户端不在线时,消息就存在数据库里,直到被这个
ID 的客户端消费掉-->
       cproperty name="clientId" value="consumerClient2"/>
       cproperty name="messageListener" ref="consumerMessageListener" />
       <!-- 消息应答方式
           Session.AUTO ACKNOWLEDGE 消息自动签收
           Session.CLIENT ACKNOWLEDGE 客户端调用 acknowledge 方法手动签收
           Session.DUPS OK ACKNOWLEDGE 不必必须签收,消息可能会重复发送
       cproperty name="sessionAcknowledgeMode" value="1"/>
   </bean>
</beans>
```

```
package com.mq.spring.topic;
import org.apache.commons.lang.builder.ToStringBuilder;
import javax.jms.JMSException;
import javax.jms.Message;
import javax.jms.MessageListener;
import javax.jms.TextMessage;
public class ConsumerMessageListener implements MessageListener{
   @Override
   public void onMessage(Message message) {
       TextMessage tm = (TextMessage) message;
       try {
           System.out.println("-----消息消费------);
           System.out.println("消息内容:\t" + tm.getText());
           System.out.println("消息 ID:\t" + tm.getJMSMessageID());
           System.out.println("消息 Destination:\t" + tm.getJMSDestination());
           System.out.println("-----更多信息------);
           System.out.println(ToStringBuilder.reflectionToString(tm));
           System.out.println("----");
       } catch (JMSException e) {
           e.printStackTrace();
    }
```

## 运行结果: