Atitit rocketMq的发送与接收

package org.chwin.firefighting.apiserver.test;  
  
import com.alibaba.fastjson.JSON;  
import com.google.common.collect.Maps;  
import com.platform.top.xiaoyu.run.service.api.finance.vo.BusinessVO;  
import lombok.SneakyThrows;  
import org.apache.rocketmq.client.producer.DefaultMQProducer;  
import org.apache.rocketmq.client.producer.SendResult;  
import org.apache.rocketmq.common.message.Message;  
import org.apache.rocketmq.remoting.common.RemotingHelper;  
  
import javax.annotation.PreDestroy;  
import java.util.Map;  
  
public class MqUtil {  
  
  
  
 @SneakyThrows  
 public static void main(String[] args) {  
  
 DefaultMQProducer producer = new DefaultMQProducer("finance\_to\_activity\_group");  
 System.*out*.println("MQ：启动ProducerTest生产者");  
 producer.setNamesrvAddr("192.168.0.4:9876");  
 producer.start();  
 MqUtil.*sendMessage*(producer,"activity\_to\_finance\_topic", "tag11", "key" + String.*valueOf*(System.*currentTimeMillis*()), "msg...");  
 }  
  
  
 @SneakyThrows  
 public static void sendMessage(DefaultMQProducer producer, String topic, String tag11, String key, String data) {  
 System.*out*.println("MQ: 生产者发送消息 :" + data);  
 Message message = null;  
  
 //转换成字符数组  
 byte[] messageBody = data.getBytes(RemotingHelper.*DEFAULT\_CHARSET*);  
 //创建消息对象 topic ,tag ,key ,body  
 message = new Message(topic, tag11, key, messageBody);  
 //同步发送消息  
 SendResult sendResult = producer.send(message);  
 //异步发送消息  
 /\*producer.send(message, new SendCallback() {  
 public void onSuccess(SendResult sendResult) {  
 System.out.println("MQ: CouponProducer生产者发送消息" + sendResult);  
 }  
  
 public void onException(Throwable throwable) {  
 System.out.println(throwable.getMessage() + throwable);  
 }  
 });\*/  
 //单向发送 只发送消息，不等待服务器响应，只发送请求不等待应答。  
 //producer.sendOneway(message);  
  
 }  
  
// @PreDestroy  
// public void stop() {  
// if (producer != null) {  
// producer.shutdown();  
// System.out.println("MQ：关闭ProducerTest生产者");  
// }  
// }  
  
  
}

接收

package org.chwin.firefighting.apiserver.test;  
  
import lombok.SneakyThrows;  
import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;  
import org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;  
import org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;  
import org.apache.rocketmq.client.consumer.listener.MessageListenerConcurrently;  
import org.apache.rocketmq.common.consumer.ConsumeFromWhere;  
import org.apache.rocketmq.common.message.MessageExt;  
  
import java.util.List;  
import java.util.function.Consumer;  
  
public class mqConstomPush {  
  
  
 @SneakyThrows  
 public static void main(String[] args) {  
 DefaultMQPushConsumer consumer = new DefaultMQPushConsumer("CID\_JODIE\_1");  
 consumer.subscribe("activity\_to\_finance\_topic", "\*");  
 consumer.setConsumeFromWhere(ConsumeFromWhere.*CONSUME\_FROM\_FIRST\_OFFSET*);  
 consumer.setInstanceName("consumer1");  
 consumer.setNamesrvAddr("192.168.0.4:9876");  
 consumer.registerMessageListener(new MessageListenerConcurrently() {  
  
 @Override  
 public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt> msgs, ConsumeConcurrentlyContext context) {  
 System.*out*.printf("%s Receive New Messages: %s %n", Thread.*currentThread*().getName(), msgs);  
 msgs.forEach(new Consumer<MessageExt>() {  
 @Override  
 public void accept(MessageExt messageExt) {  
 System.*out*.println( new String( messageExt.getBody()) );  
 }  
 });  
 return ConsumeConcurrentlyStatus.*CONSUME\_SUCCESS*;  
 }  
 });  
 consumer.start();  
 }  
}