链滴

rabbitMQ 路由模式 - 消费者筛选订阅

作者：hanzanr123
原文链接：https://ld246.com/article/1585964559914
来源网站：链滴

生产者代码

```
import java.util.HashMap;
import java.util.Map;

import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;
/**
 * mq路由-生产者
 * @author pengzai
 *
 */
public class RoutingSendUtil {

    private static final String EXCHANGE_NAME = "direct_pengzai";        //交换机名称
    private static final String EXCHANGE_TYPE = "direct";            //交换类型
    private static final String CHARSET = "UTF-8";                   //字符集
    private static final Integer X_MESSAGE_TTL = 180*1000;           //消息超时(3分钟)
    private static final Integer X_EXPIRES = 600*1000;               //队列超时(10分钟)
    private static final Integer X_MAX_LENGTH = 1;                   //长度限制(1条)

    private static ConnectionFactory factory = new ConnectionFactory();
    static{
        factory.setHost("127.0.0.1");    //mq服务ip
        factory.setPort(5672);               //mq服务端口
        factory.setVirtualHost("/pengzai");    //mq虚拟主机
        factory.setUsername("admin");        //mq账户
        factory.setPassword("admin");        //mq密码
    }

    /**
     * 发送消息
     * @param sn  --设备sn号
     */
    public static void execute(String sn) {

        String queueName = sn;    //队列名称

        String routingKey = sn;    //路由key

        Connection connection = null;
        Channel channel = null;
        try {
            // 建立TCP连接
            connection = factory.newConnection();
            // 在TCP连接的基础上创建通道
            channel = connection.createChannel();
            // 声明一个direct交换机
            channel.exchangeDeclare(EXCHANGE_NAME, EXCHANGE_TYPE);
```

```
            //创建(声明)队列
            Map<String,Object> params = new HashMap<String, Object>();
            params.put("x-message-ttl", X_MESSAGE_TTL);    //消息超时
            params.put("x-expires", X_EXPIRES);            //队列超时
            params.put("x-max-length", X_MAX_LENGTH);    //长度限制
            channel.queueDeclare(queueName, false, false, true, params);

            //绑定 队列-交换机-路由
            channel.queueBind(queueName, EXCHANGE_NAME, routingKey);

            String message = "RoutingSend-" + System.currentTimeMillis();
            // 发送消息，并配置消息的路由键
            channel.basicPublish(EXCHANGE_NAME, routingKey, null, message.getBytes(CHARSET)
;
            System.out.println("发送成功");
        }catch (Exception e){
            e.printStackTrace();
        }finally {
            try {
                // 空值判断，为了代码简洁略
                channel.close();
                connection.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }

    public static void main(String[] args) {
        execute("1212415450");
    }
}
```

消费者代码

```
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;
import com.rabbitmq.client.QueueingConsumer;
import com.rabbitmq.client.QueueingConsumer.Delivery;
/**
 * mq路由-消费者
 * @author pengzai
 *
 */
public class RoutingRecvUtil {
    private static final String EXCHANGE_NAME = "direct_pengzai";        //交换机名称
    private static final String EXCHANGE_TYPE = "direct";            //交换类型
```

```
private static final String CHARSET = "UTF-8";                  //字符集
private static final Integer X_MESSAGE_TTL = 180*1000;          //消息超时(3分钟)
private static final Integer X_EXPIRES = 600*1000;              //队列超时(10分钟)
private static final Integer X_MAX_LENGTH = 1;                  //长度限制(1条)

private static ConnectionFactory factory = new ConnectionFactory();
static{
    factory.setHost("127.0.0.1");    //mq服务ip
    factory.setPort(5672);           //mq服务端口
    factory.setVirtualHost("/pengzai");    //mq虚拟主机
    factory.setUsername("pengzai");          //mq账户
    factory.setPassword("pengzai");          //mq密码
}

/**
 * 接收mq消息
 * @param sn  --设备sn号
 */
public static void execute(String sn){

    String queueName = sn;    //队列名称

    String routingKey = sn;    //路由key

    Connection connection = null;
    Channel channel = null;
    try {
        // 建立TCP连接
        connection = factory.newConnection();
        // 在TCP连接的基础上创建通道
        channel = connection.createChannel();
        // 声明一个direct交换机
        channel.exchangeDeclare(EXCHANGE_NAME, EXCHANGE_TYPE);

        //创建(声明)队列
        Map<String,Object> params = new HashMap<String, Object>();
        params.put("x-message-ttl", X_MESSAGE_TTL);    //消息超时
        params.put("x-expires", X_EXPIRES);            //队列超时
        params.put("x-max-length", X_MAX_LENGTH);    //长度限制
        channel.queueDeclare(queueName, false, false, true, params);

        // 绑定队列,路由
        channel.queueBind(queueName, EXCHANGE_NAME, routingKey);

        // 定义消息的回调处理类
        QueueingConsumer queueingConsumer = new QueueingConsumer(channel);

        // 接收消息
        channel.basicConsume(queueName, true, queueingConsumer);

        System.out.println("等待接收消息.....");
        while (true) {
            Delivery delivery = queueingConsumer.nextDelivery();
            String msg = new String(delivery.getBody(),CHARSET);
```

```
                System.out.println("消息内容:"+msg);
            }
        }catch (Exception e){
            e.printStackTrace();
        }finally {
            try {
                channel.close();
                connection.close();
            } catch (Exception x) {

            }
        }
    }

    public static void main(String[] args) throws IOException {
        execute("1212415450");
    }
}
```