

☐ 皮肤

最后修改于: 2021/03/06 00:05

SIMPLE值经测试有两种效果，其一效果和CORRELATED值一样会触发回调方法，其二在发布消息成功后使用rabbitTemplate调用waitForConfirms或waitForConfirmsOrDie方法等待broker节点返回发送结果，根据返回结果来判定下一步的逻辑，要注意的点是waitForConfirmsOrDie方法如果返回false则会关闭channel，则接下来无法发送消息到broker;

```
1. # 服务端点
2. server:
3.     port: 8080
4.
5. # 配置rabbitmq服务
6. spring:
7.     rabbitmq:
8.         username: admin
9.         password: admin
10.        virtual-host: /
11.        host: 47.104.141.27
12.        port: 5672
13.        publisher-confirm-type: correlated
```

```

1. package com.xuexiangban.rabbitmq.springbootorderrabbitmqproducer.callback;
2.
3. import org.springframework.amqp.rabbit.connection.CorrelationData;
4. import org.springframework.amqp.rabbit.core.RabbitTemplate;
5. import org.springframework.stereotype.Component;
6.
7. /**
8.  * @description:
9.  * @author: xuke
10.  * @time: 2021/3/5 23:25
11.  */
12. public class MessageConfirmCallback implements RabbitTemplate.ConfirmCallback {
13.
14.     @Override
15.     public void confirm(CorrelationData correlationData, boolean ack, String cause) {
16.         if(ack){
17.             System.out.println("消息确认成功!!!!");
18.         }else{
19.             System.out.println("消息确认失败!!!!");
20.         }
21.     }
22. }

```

```
1. /**
2.     * @Author xuke
3.     * @Description 模拟用户购买商品下单的业务
4.     * @Date 22:26 2021/3/5
5.     * @Param [userId, productId, num]
6.     * @return void
7.     **/
8. public void makeOrderTopic(String userId,String productId,int num){
9.     // 1: 根据商品id查询库存是否充足
10.    // 2: 保存订单
11.    String orderId = UUID.randomUUID().toString();
12.    System.out.println("保存订单成功: id是: " + orderId);
13.    // 3: 发送消息
14.
15.    //com.# duanxin
16.    //#.email.* email
17.    //#.sms.# sms
18.    // 设置消息确认机制
19.    rabbitTemplate.setConfirmCallback(new MessageConfirmCallback());
20.    rabbitTemplate.convertAndSend("topic_order_ex","com.email.sms.xxx",orderId);
21. }
```



[关于我们](#) | [加入我们](#) | [联系我们](#) | [帮助中心](#)

Copyright © 广东学相伴网络科技有限公司 [粤ICP备 - 2020109190号](#)

