Rocketmq使用文档

1. 添加依赖

<**dependency**>  
 <**groupId**>com.rongxintianxia.common</**groupId**>  
 <**artifactId**>rxtx-common-rocketmq</**artifactId**>  
 <**version**>0.0.1</**version**>  
</**dependency**>

二, 修改application.yml

修改mybatis配置 classpath后跟\*

*# Mybatis配置***mybatis**:  
 **mapperLocations**: classpath\*:sqlmap/\*\*/\*.xml

添加

**rongxin**:  
 **rocketmq**:  
  **namesrvAddr**: 192.168.3.22:9876;192.168.3.19:9876  
 **producerMaxRetryCount**: 10  
 **consumeMaxRetryCount**: 10

其中namesrvAddr为当前开发环境rocketmq的namesrvAdd，测试与生产另行提供

**producerMaxRetryCount** 为发送失败后最大重新发送次数，可根据需求自行定义，暂定为10

**consumeMaxRetryCount** 为消息消费失败后最大重新消费次数，可根据需求自定义，暂定为10

三, 项目启动类注解修改：

mapperscan注解修改为：@MapperScan(**"com.rongxintianxia.\*\*.dao.mapper"**)

需要发送消息的话，新增注解：@EnableRocketMqProducer

需要接收消息的话，新增注解：@EnableRocketMqConsumer

四， 发送消息：

1.在需要发送消息的类引入bean:

@Autowired  
**private** IMqProducer **mqProducerImpl**;

2. 在需要发送消息的类实现CommandLineRunner 接口,在接口定义的run方法里，启动producer:

**mqProducerImpl**.start(MqChannel.***MQ\_CHANNEL\_TEST***.getProducerGroup());

3.发送消息：

**mqProducerImpl**.sendMsg(MqChannel.***MQ\_CHANNEL\_TEST***.getTopic(),, msgData);

4.在2和3中produerGroupName, topicName在rxtx-common-rocketmq下枚举类MqChannel里根据需求添加，不要各个项目自定义,以防止消息收发错乱。

五, 接收消息:

1.在需要接收消息的类里引入bean：

@Autowired  
private IMqConsumer mqConsumerImpl;

2．在需要发送消息的类实现CommandLineRunner 接口,在接口定义的run方法里，启动consumer:

**mqConsumerImpl**.start(MqChannel.***MQ\_CHANNEL\_TEST***.getTopic(), MqChannel.***MQ\_CHANNEL\_TEST***.getConsumerGroup(), **new** IConsumerListener() {  
 @Override  
 **public** String apply(String msg) {  
 System.***out***.println(**"收到消息"** + msg);  
 *//处理收到的消息* **return** RetCode.***SUCC***.getCode();  
 }  
});

其中mqConsumerImpl.start 方法第一个参数为mqChannel中定义的topic的名称，第二个为topic对应的consumerGroup，IConsumerListener回调中msg为收到的消息。在回调中处理消息。返回RetCode.***SUCC***.getCode()表示消费成功，返回其它表示消息消费失败，会再次重复消费该消息。