各位好：

   rocketmq3.x 服务已经在生产平稳运行了3+年，目前官方已经不再对3.x维护了，接下来将会对生产3.x平滑过渡到4.x，

   在这个平滑转换过程中对业务系统生产服务不会造成影响，3.x服务与4.x服务会同时并行(并行运行的长短,会根据各个组的情况而定)，平滑迁移方案针对各个组的业务情况，

   我这边会协助制定平稳的迁移方案；

   此邮件主要针对4.3的集群安装部署先期知会各位，给予各位在(测试环境、研发环境)充分的时间准备;

   在安装部署过程中有任何问题及时联系我，感谢各位配合。

   测试环境，请@藏老板安排时间先期准备一下，辛苦。

   另：客户端适用包我这边会单独提供。先期

**各位好**：

**=>同步发送消息;**

**=>异步发送消息**

**=>广播发送消息;**

**=>有序发送与消费消息；**

**=>发送延迟消息；**

**=>消息tag与key支持；**

**=>自动序列化与反序列化消息体；**

**=>消息的实际消费方IP追溯；**

**=>发送事务消息；**

**\* rocketmq升级是架构调整的一部分,接下来架构调整已经到了加速期， 希望各个业务端尽可能在【3周之内】升级完毕,**

**目前已经加速在融合微服务生态架构：@springboot@springcloud@dubbo@rocketmq@nacos@scheduleX@gateway@http2@ACL@docker@k8s**

     鉴于是大版本升级(**rocket3.2.4 to rocketmq4.3.2**) 为了平滑、稳定过渡，此次升级分了三步走(服务端进行了两次升级、客户端需要做一次升级)

     服务端的两次升级已经完毕，目前平稳运行；

**目前通知各个业务线客户端进行升级操作，客户端升级操作具体步骤如下**：(业务客户端---生产者与消费者各自安排时间进行升级，不需要强制两个业务端同时去升级)

     基础环境: **JDK1.8+**

**# 【升级方案一】：简单单一，只需要变更一下pom rocketmq版本与代码里面的相关mq引用的包路径**

**1.引入POM:**

**<dependency>**

**<groupId>org.apache.rocketmq</groupId>**

**<artifactId>rocketmq-client</artifactId>**

**<version>4.3.2</version>**

**</dependency>**

**2. rocketmq3.x to rocketmq4.3.2**

**引用的package名做变更**： com.alibaba.rocketmq.\*  变更为 org.apache.rocketmq.\*

**引用服务端的相关配置保持原有不变；**

**# 【升级方案二】：引用spring-boot-rocketmqmq-starter；**

    (此方案优势，业务系统逻辑与rocketmq逻辑解耦，直接使用client包提供的接口就能实现消息的发送与消费，简化了在业务系统上耦合过多的代码)

    1. 引入maven依赖：

**<dependency>**

**<groupId>tech.yooli.rocketmq</groupId>**

**<artifactId>spring-boot-rocketmq-starter</artifactId>**

**<version>0.2.0.RELEASE</version>**

**</dependency>**

**#消息生产者示例请参考: (重点关注标红)**

##### **程序入口添加注解开启自动装配**

在springboot应用主入口添加@EnableMQConfiguration注解开启自动装配：

@EnableRocketMQConfiguration

**public** **class** SpringBootRocketMQStarerApplication {

**public** **static** **void** main(String[] args) {

     SpringApplication.*run*(SpringBootRocketMQStarerApplication.**class**, args);

    }

}

**# 配置信息:(application.yml)**

spring:

  yooli:

    rocketmq:

      name-server-address: IP1:port;IP2:port

producer-group: spring-boot-rocketmq-starter-producer

**# 创建发送方**：

@RocketMQProducer

**public** **class** RocketMQProducerDemo **extends** AbstractRocketMQProducer { }

**# 发送消息:**

@Autowired

**private** RocketMQProducerDemo demoProducer;

...

demoProducer.sysncSend(msg);

**# 构建消息体:**

  MessageBuilder.*of*("message contents")

               .topic("tech\_yooli\_springboot\_topic")

               .build();

2.消息消费者示例请参考:

**创建消费方**：

*@Slf4j*

@RocketMQConsumer(topic = "${yooli.cusumer.topic}", consumerGroup ="yooli\_consumer\_group")

**public** **class** RocketConsumerDemo **extends** AbstractRocketMQPushConsumer{

    @Override

**public** **boolean** process(Object message, Map extMap) {

***log***.info("消费端订阅的消息内容:{}", message);

**return** **true**;

    }

}

这个升级过程中，感谢测试同学与运维同学的大力支持。