# PPM MQ架构设计

MQ: Spring AMQP+Rabbit MQ

IM：极光IM

Mail: Spring boot starter Mail

# 1 引言

## 背景

为了提高系统并发能力及高可用性，使用队列机制可以帮助系统提高响应速度，系统稳定性。生产、消费者模式可以将串行的处理异步化、解耦、有效消除峰值。且更有利于构建分布式集群环境。

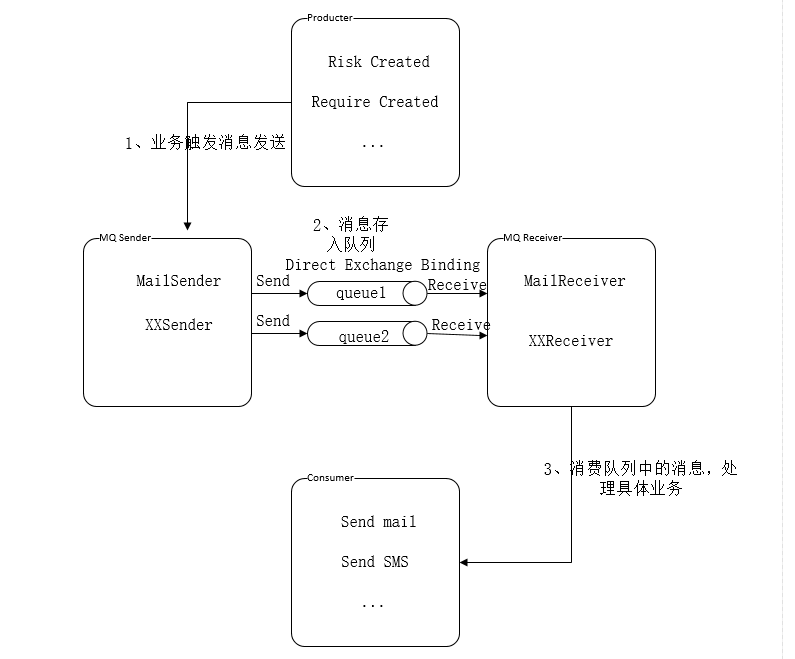
## 技术栈

Spring AMQP

Rabbit MQ

Velocity （模板技术）

# 2 逻辑架构图



## 名词解释

Producter: 业务触发点，生产消息，如：创建风险、需求之后希望发送邮件给相关处理人，调用MQ Sender服务（Restful）发送消息。

MQ Sender：消息发送者，将消息发送到绑定的队列。

MQ Receiver：消息接受者，接受监听队列消息，并调用Consumer服务（Restful）处理相关业务。

Consumer：消费消息，处理具体业务。

# 3 架构解析

## 组成

消息队列分两个部分组成：消息发送者（mq-sender）、消息接受者(mq-receiver)

## 技术栈

Spring Boot+Spring AMQP+Rabbit MQ

消息发送者（mq-sender）、消息接受者(mq-receiver)分别都是一个Spring Boot工程，可单独运行。

## mq-sender

@Component

**public** **class** MailSender {

**private** **static** **final** String ***MAIL\_ROUTE\_KEY*** = "mail.route.key";

**private** **static** **final** String ***MAIL\_QUEUE*** = "mailQueue";

@Autowired

**private** RabbitTemplate rabbitTemplate;

@Bean

**public** Queue mailQueue() {

**return** **new** Queue(***MAIL\_QUEUE***, **true**);

}

@Bean

**public** Binding myExchangeBinding(@Qualifier("directExchange") DirectExchange directExchange,

@Qualifier(***MAIL\_QUEUE***) Queue queue) {

**return** BindingBuilder.*bind*(queue).to(directExchange).with(***MAIL\_ROUTE\_KEY***);

}

**public** **void** send(String content) {

content += **new** Date();

System.***out***.println("Sender : " + content);

**this**.rabbitTemplate.convertAndSend("directExchange",***MAIL\_ROUTE\_KEY***,content);

}

}

## mq-receiver

@Component

@RabbitListener(queues = "mailQueue",containerFactory="rabbitListenerContainerFactory")

**public** **class** MailReceiver {

**private** **static** **final** String ***MAIL\_ROUTE\_KEY*** = "mail.route.key";

**private** **static** **final** String ***MAIL\_QUEUE*** = "mailQueue";

@Autowired

**private** MailService mailService;

@Bean

**public** Queue mailQueue() {

**return** **new** Queue(***MAIL\_QUEUE***, **true**);

}

@Bean

**public** Binding myExchangeBinding(@Qualifier("directExchange") DirectExchange directExchange,

@Qualifier(***MAIL\_QUEUE***) Queue queue) {

**return** BindingBuilder.*bind*(queue).to(directExchange).with(***MAIL\_ROUTE\_KEY***);

}

@RabbitHandler

**public** **void** process(String msg) {

System.***out***.println("Receiver : " + msg);

mailService.testSimpleTextMail();

}

}

# 4 邮件服务

Path：<http://localhost:8085/ppm-mq-sender/msg/mail>

Method：POST

Request body：

{

"subject": "这是测试邮件2", //邮件主题

"from": "eleadrd@aliyun.com", //发件人

"to": [ //收件人（多个）

"liwei@e-lead.cn"

],

"cc": [ //抄送人（多个）

"liwei@e-lead.cn"

],

"textContent": "这是测试邮件内容。", //邮件内容

"type":"Html" // 邮件类型：Text、Html

}