

Please visit website: <http://cxyroad.com>

.PriorityMessageTest|INFO|59|Basic.Return 返回结果: mandatory test

...

也可以通过 `RabbitMQ` 管理界面, 看 `Exchange` 概况来确定消息是否投递过

作为拓展, 给你们留两个问题

- > 1. mandatory 设置为 true 的同时, 不添加监听器 ReturnListener, 会是什么结果
- > 2. mandatory 设置为 false 的同时, 添加监听器 ReturnListener, 又会是什么结果

### immediate

`immediate` 参数用于设置消息是否立即发送给消费者, 默认值是 `false`

当 `immediate` 参数设置为 `true` 时, 如果消息路由到队列时发现队列上并没有任何消费者, 那么该消息不会存入队列中, 当与路由键匹配的所有队列都没有消费者时, 该消息会通过 `Basic.Return` 返回至生产者

> immediate 为 true , 消息路由到匹配的队列时

>

>

- > 1. 部分队列有消费者, 有消费者的队列会立即将消息投递给消费者, 没有消费者的队列会丢弃该消息
- > 2. 全部队列都没有消费者, 则将该消息返回给生产者

执行如下代码

```
!(https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/31a9eb56d67249afa7a5901ebfec3ce1~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=893&h=238&s=37411&e=png&b=fefdfd)
```

你会发现报错

...

```
2024-06-01 16:16:06|AMQP Connection  
10.5.108.226:5672|org.springframework.amqp.rabbit.connection.Caching  
ConnectionFactory|ERROR|1575|Channel shutdown: connection error;  
protocol method: #method<connection.close>(reply-code=540, reply-  
text=NOT_IMPLEMENTED - immediate=true, class-id=60, method-  
id=40)
```

...

这是因为从 `RabbitMQ 3.0` 版本开始去掉了对 `immediate` 参数的支持，对此官方解释如下

> `immediate` 参数会影响镜像队列的性能，增加了代码复杂性，建议采用 `TTL` 和 `DLX` 替代 `immediate`

概括来讲，`mandatory` 针对的是消息能否路由到至少一个队列中，否则将消息返回给生产者。`immediate` 针对的是消息能否立即投递给消费者，否则将消息直接返回给生产者，不用将消息存入队列而等待消费者

### ### Alternate Exchange

生产者在发送消息时，如果不设置 `mandatory` 参数（或设置为 `false`），那么消息在未被路由的情况下会丢失；如果设置了 `mandatory`（且设置成 `true`），那么需要添加对应的 `ReturnListener` 逻辑，生产者的代码会变得复杂。如果既不想增加生产者的复杂，又不想消息丢失，那么就可以使用备份交换器（`Alternate Exchange`），将未被路由的消息存储在 `RabbitMQ` 中，在需要的时候再去处理这些消息

实现代码如下

```

```

执行如下测试代码

```
 消息通过 `com.qsl.normal.exchange`，经路由键 `123` 未匹配到任何队列，此时消息就会发送给 `com.qsl.normal.exchange` 的备份交换机 `com.qsl.alternate.exchange`，因为备份交换器的类型是 `fanout`，所以消息会被路由到 `com.qsl.alternate.exchange` 绑定的所有队列上，目前只有一个队列 `com.qsl.unrouted.queue`，所以消息最终来到 `com.qsl.unrouted.queue`，消息流转如下

 消费者端配置

  
消息发送

  
输出日志如下

  
实际应用中，可以根据延迟时长给延迟队列划分多个等级，例如

  
目前 `RabbitMQ` 提供了另外的方式来实现 `延迟队列`

> [github.com/rabbitmq/ra...](http://cxyroad.com/"https://github.com/rabbitmq/rabbitmq-delayed-message-exchange")

感兴趣的可以去看看

## 总结

--

1. 示例代码: [spring-boot-rabbitmq](http://cxyroad.com/"https://gitee.com/youzhibing/qs1-project/tree/master/spring-boot-rabbitmq")
2. mandatory 与 immediate

`mandatory` 针对的是消息能否路由到至少一个队列中，否则将消息返回给生

## 产者

`immediate` 针对的是消息能否立即投递给消费者，否则将消息直接返回给生产者，不用将消息存入队列而等待消费者

`RabbitMQ 3.0` 版本开始去掉了对`immediate`参数的支持，可以用`DLX`和`TTL`来代替

### 3. 过期时长

消息的过期时长有两种设置方式：队列的参数`x-message-ttl`和消息的参数`expiration`

队列也可以设置过期时长，该时长内队列一直处于未使用状态则会被删除；通过队列参数`x-expires`来设置

### 4. 死信队列

绑定到死信交换器（`DLX`）上的队列就是死信队列

`DLX`能够保证异常的情况下消息不会丢失，后续通过分析死信队列中的消息，可以改善和优化系统

### 5. 延迟队列

目前来讲，实现延迟队列的方式有两种

#### 1. `DLX`与`TTL`

2. [rabbitmq-delayed-message-exchange](<http://cxyroad.com/>  
”<https://github.com/rabbitmq/rabbitmq-delayed-message-exchange>”)

## 参考

--

《RabbitMQ实战指南》

原文链接: <https://juejin.cn/post/7375370443732549669>