Rabbitmq笔记

### rabbitmq安装（Ubuntu）

#### 安装erlang

由于rabbitMq需要erlang语言的支持，在安装rabbitMq之前需要安装erlang

执行 sudo apt-get install erlang-nox

#### 安装Rabbitmq

更新源 sudo apt-get update

安装 sudo apt-get install rabbitmq-server

......以下是rabbitmq的web界面管理工具

#### 查看rabbitmq的插件列表: sudo rabbitmq-plugins list

启用RabbitMQ WEB管理界面: rabbitmq-plugins enable rabbitmq\_management

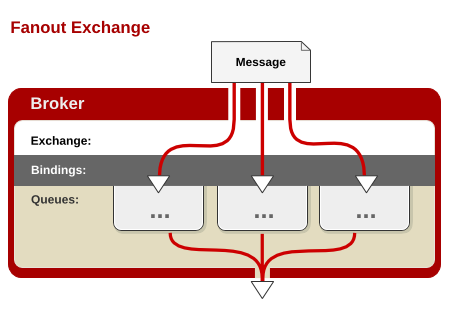
访问管理界面： http://192.168.20.151:15672 默认用户名/密码：guest/guest

#### 服务重启：systemctl restart rabbitmq

### RabbitMQ三种Exchange模式——订阅、路由、通配符模式

#### 订阅模式(Fanout Exchange)

一个生产者，多个消费者，每一个消费者都有自己的一个队列，生产者没有将消息直接发送到队列，而是发送到了交换机，每个队列绑定交换机，生产者发送的消息经过交换机，到达队列，实现一个消息被多个消费者获取的目的。需要注意的是，如果将消息发送到一个没有队列绑定的exchange上面，那么该消息将会丢失，这是因为在rabbitMQ中exchange不具备存储消息的能力，只有队列具备存储消息的能力。



##### Java代码示例（使用的spring包装的AmqpTemplate）

i: 生产者发送消息到exchange（fanout模式）

Template's template = (AmqpTemplate) factory.getBean("rabbitTemplate");

template.convertAndSend("amq.fanout", **null**, userLogString);

ii: 消费者接收消息

@Component

**public** **class** ReceiverListener {

**private** **static** **final** Logger ***logger*** = LoggerFactory.*getLogger*(ReceiverListener.**class**);

@RabbitListener(queues = "track") // 监听器监听访问记录

**public** **void** userTrack(String userLog) {

// 监听到消息

***logger***.info("track接收到的消息"+userLog);

}

@RabbitListener(queues = "trace") // 监听器监听访问记录

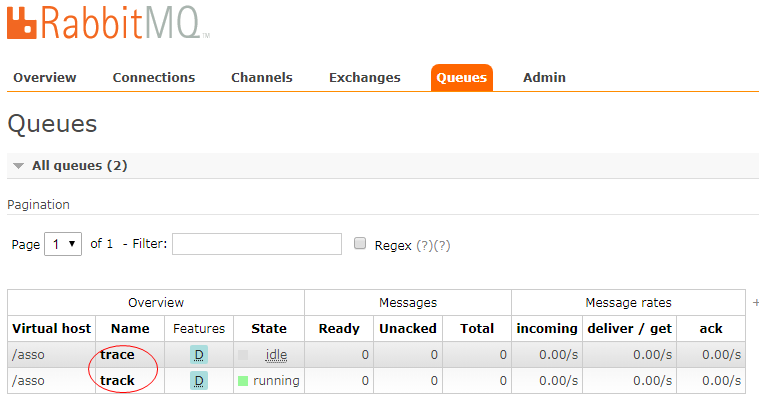
**public** **void** usertestTrack(String userLog) {

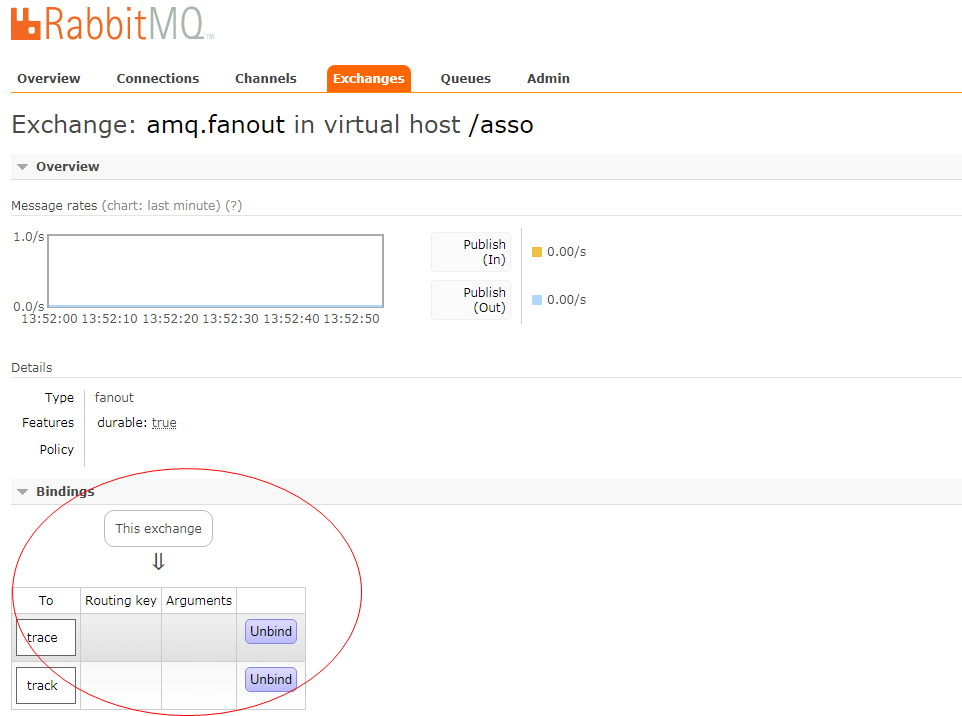
***logger***.info("trace接收到的消息"+userLog);

}

}

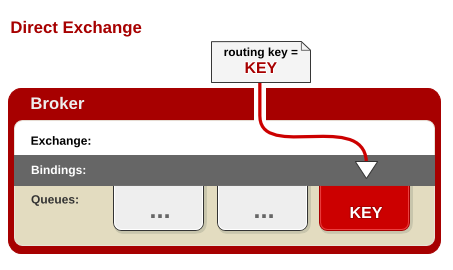
要想上述消息发送成功还需要去管理界面添加相应的Queues，并把要接收消息的quenues绑定到交换机





#### 路由模式(Direct Exchange)

这种模式添加了一个路由键，生产者发布消息的时候添加路由键，消费者绑定队列到交换机时添加键值，这样就可以接收到需要接收的消息。



##### Java代码示例

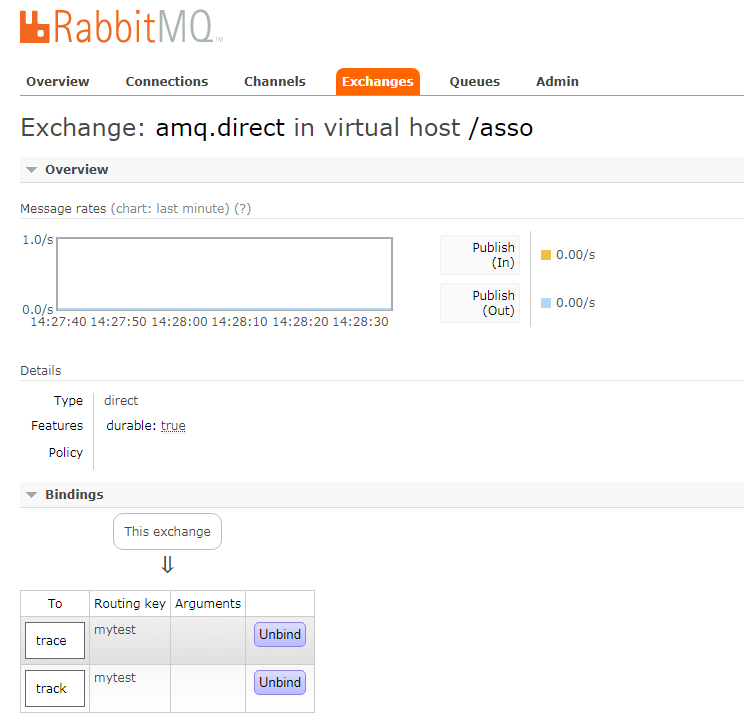
i: 生产者发送消息到exchange（direct模式）

template.convertAndSend("amq.direct", "mytest",userLogString);

ii: 消费者接收消息

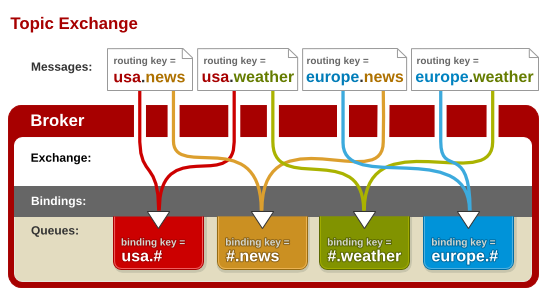
消费者代码与上述订阅模式一致

管理界面需要进行绑定（注意绑定的路由名称要和生产者发送的路由名称一致否则消息将丢失）



#### 通配符模式（Topic Exchange）

基本思想和路由模式是一样的，只不过路由键支持模糊匹配，符号“#”匹配一个或多个词，符号“\*”匹配不多不少一个词（单词之间用“.”进行隔开 ）



Java*代码示例*

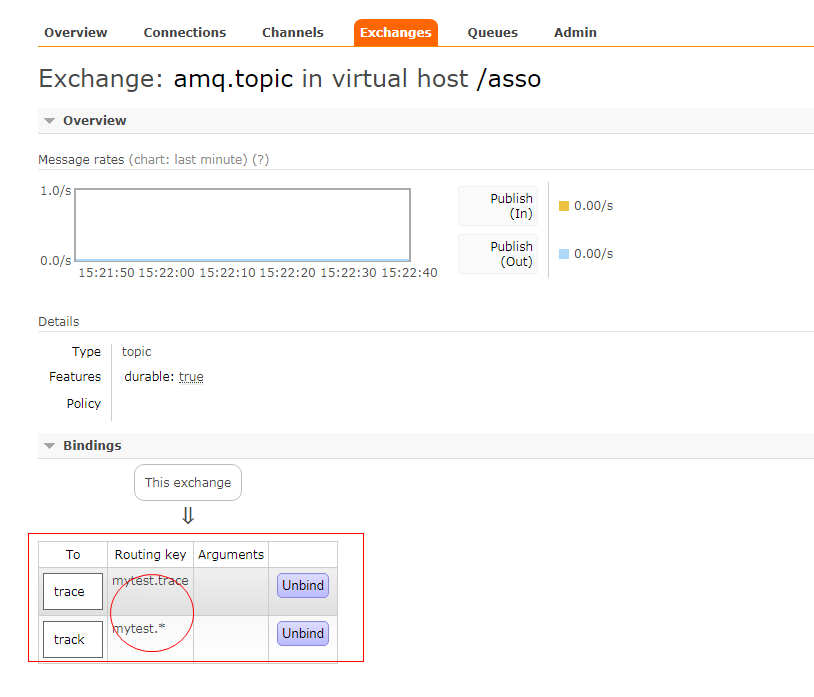
i: 生产者发送消息到exchange（direct模式）

template.convertAndSend("amq.topic", "mytest.",userLogString);

ii: 消费者接收消息

消费者代码与上述订阅模式一致

*管理界面相关配置*



#### 小结

这三种模式，均可以实现一个消息被多个消费者获取。.fanout这种模式没有加入路由器，队列与exchange绑定后，就会接收到所有的消息，其余两种增加了路由键，并且第三种增加通配符，更加便利。当然了，管理界面的那些队列绑定到交换机以及绑定时输入的路由配置都可以在代码中进行。

### 轮询模式

Rabbitmq的queues具有轮询功能，即queue下的消费者消费依次平均消费消息。（假如有六个消息，三个消费者，这六个消息依次标为123456，消费者为abc，然后这些消息会按照1->a,2->b,3->c,4->a,5->b,6->c依次消费，即使abc服务处理性能不一致）。这时就不需要昂贵的负载均衡。下图就是消费者消费信息，（a对应1，其中a服务处理速度慢，但可以看出是平均分发的）

