## 定义

把消息按照产生的次序加入队列，而由另外的处理程序/模块将其从队列中取出，并加以处理；从而形成了一个基本的消息队列。使用消息队列可以很好地将任务以异步的方式进行处理，或者进行数据传送和存储等。例如，当你频繁地向数据库中插入数据、频繁地向搜索引擎提交数据，就可采取消息队列来异步插入。另外，还可以将较慢/较复杂的处理逻辑、有并发数量限制的处理逻辑，通过消息队列放在后台处理。

## 使用场景

速派消息通知：异步实现发手机短信、发邮件、发App推送、发站内信

## 服务器搭建

* 搭建步骤

见<http://www.cnblogs.com/yangecnu/p/4227535.html>

* 配置常量

ERLANG\_HOME C:\Program Files (x86)\erl5.8.4

Path： ;%RABBITMQ\_SERVER%\sbin

* 密令：

启动监控管理器：rabbitmq-plugins enable rabbitmq\_management

关闭监控管理器：rabbitmq-plugins disable rabbitmq\_management

启动rabbitmq：rabbitmq-service start

关闭rabbitmq：rabbitmq-service stop

查看所有的队列：rabbitmqctl list\_queues

清除所有的队列：rabbitmqctl reset

关闭应用：rabbitmqctl stop\_app

启动应用：rabbitmqctl start\_app

添加用户：rabbitmqctl add\_user username password

分配角色：rabbitmqctl set\_user\_tags username administrator

新增虚拟主机：rabbitmqctl add\_vhost vhost\_name

将新虚拟主机授权给新用户：rabbitmqctl set\_permissions -p vhost\_name username '.\*' '.\*' '.\*'

## .Net中使用

1. 消息持久化要做到以下三点：

* 消息持久化：保证消息持久化,把消息的“投递模式”设置为2，

properties.DeliveryMode = 2;

properties.Persistent = true;

* 发送到持久化的交换器

bool durable = true;

channel.ExchangeDeclare(exchangeName, exchangeType, durable, false, null);

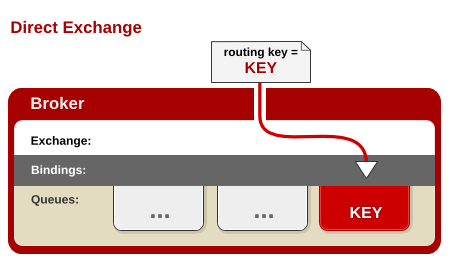
* 发送到持久化的队列

bool durable = true;

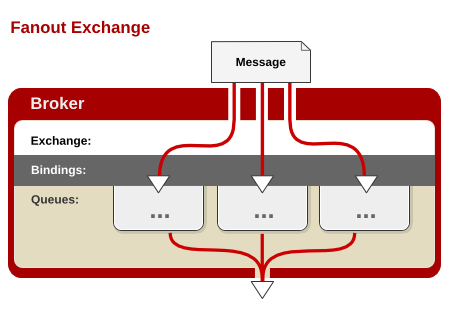
channel.QueueDeclare(queueName, durable, false, false, null);

1. 路由方式：(http://www.cnblogs.com/knowledgesea/p/5296008.html)

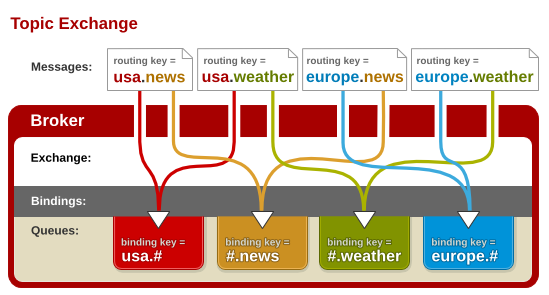
* Direct Exchange – 处理路由键。需要将一个队列绑定到交换机上，要求该消息与一个特定的路由键完全匹配。这是一个完整的匹配。如果一个队列绑定到该交换机上要求路由键 “dog”，则只有被标记为“dog”的消息才被转发，不会转发dog.puppy，也不会转发dog.guard，只会转发dog。



* **Fanout Exchange** – 不处理路由键。你只需要简单的将队列绑定到交换机上。一个发送到交换机的消息都会被转发到与该交换机绑定的所有队列上。很像子网广播，每台子网内的主机都获得了一份复制的消息。Fanout交换机转发消息是最快的。



* **Topic Exchange** – 将路由键和某模式进行匹配。此时队列需要绑定要一个模式上。符号“#”匹配一个或多个词，符号“\*”匹配不多不少一个词。因此“audit.#”能够匹配到“audit.irs.corporate”，但是“audit.\*” 只会匹配到“audit.irs”。我在RedHat的朋友做了一张不错的图，来表明topic交换机是如何工作的。



1. 虚拟主机vhost

RabbitMQ服务器权限控制以vhost为单位，每个vhost有自己的队列、路由，不同的vhost里面队列，路由可以重名，每一个vhost实例提供逻辑上的分离，所以不同的项目可以使用不同的vhost便于管理。