# RabbitMQ

RabbitMQ 是一个基于 AMQP 协议实现的开源消息队列系统，它提供了可靠的消息传递机制，支持多种消息传递模式和高度灵活的配置选项。

## 1. 什么是消息队列（Message Queue）

消息队列是一种在应用程序之间传递消息的通信方式。它通常用于解耦生产者和消费者，使得系统可以更灵活、可靠地处理消息传递过程。

## 2. 为什么需要消息队列

解耦性: 可以降低系统组件之间的耦合度，提高系统的灵活性和可维护性。

异步通信: 生产者和消费者可以异步地进行通信，提高系统整体性能和响应速度。

削峰：将高并发时的同步方为变为串行访问达到一定量的限流，利于数据库的操作。

消息持久化: 消息队列可以将消息持久化，确保即使在系统故障或重启后消息不丢失。

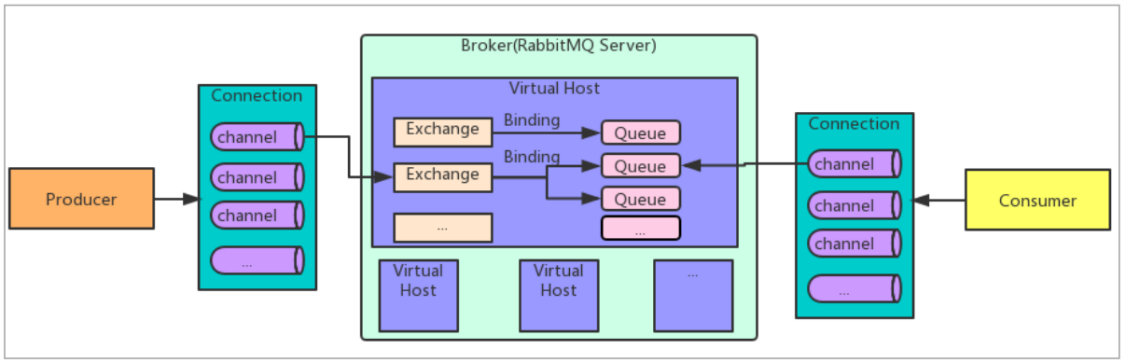
消息分发: 可以实现消息的广播和点对点的分发，满足不同的消息传递需求。

## 3. AMQP（Advanced Message Queuing Protocol）简介

AMQP，即 Advanced Message Queuing Protocol（高级消息队列协议），是一个网络协议，是应用层协议的一个开放标准，为面向消息的中间件设计。基于此协议的客户端与消息中间件可传递消息，并不受客户端/中间件不同产品，不同的开发语言等条件的限制。

AMQP这个过程就是：由生产者发布消息到交换机，交换机再通过路由规则将消息发送到不同的队列中去存储，然后消费者从队列中监听拿走对应的消息来消费。

## RabbitMQ 的核心概念



Broker（服务节点）：接收和分发消息的应用，RabbitMQ Server就是 Message Broker。

Virtual host（虚拟机）：出于多租户和安全因素设计的，把 AMQP 的基本组件划分到一个虚拟的分组中，类似于网络中的 namespace 概念。当多个不同的用户使用同一个 RabbitMQ server 提供的服务时，可以划分出多个vhost，每个用户在自己的 vhost 创建 exchange／queue 等

Exchange（交换机）: 用于接收生产者发送的消息，并根据路由规则将消息路由到一个或多个队列。

Queue（队列）: 用于存储消息，消费者可以从队列中获取消息进行处理。

Channel（通道）：如果每一次访问 RabbitMQ 都建立一个 Connection，在消息量大的时候建立 TCP Connection的开销将是巨大的，效率也较低。Channel 是在 connection 内部建立的逻辑连接，如果应用程序支持多线程，通常每个thread创建单独的 channel 进行通讯，AMQP method 包含了channel id 帮助客户端和message broker 识别 channel，所以 channel 之间是完全隔离的。Channel 作为轻量级的 Connection 极大减少了操作系统建立 TCP connection 的开销

Routing Key（路由键）: 生产者将消息发送到 Exchange 时指定的键，Exchange 根据路由规则将消息路由到相应的队列。

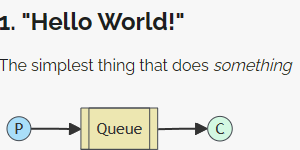
Binding（绑定）: 定义了 Exchange 和 Queue 之间的关联关系，指定了消息的路由规则。

Producer（生产者）: 发布消息到 Exchange。

Consumer（消费者）: 从队列中获取消息并进行处理。

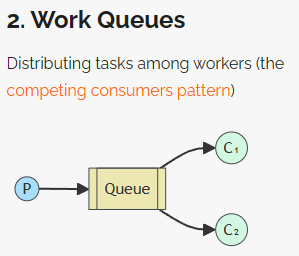
## RabbitMQ 工作模式

（1）简单模式（hello world）



一个生产者、一个消费者，一对一发送消息，主要用于学习和演示。该模式没有消息持久化、没有消息确认机制、没有消息队列的持久化，如果生产者发送消息的速度大于消费者消费消息的速度，消费者可能会丢失消息。

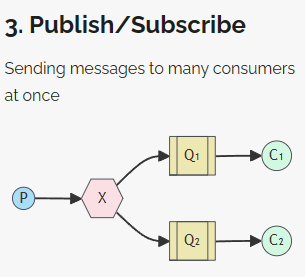
（2）工作队列模式（Work Queues）



一个生产者、多个消费者，一对多发送消息。当需要多个消费者共同处理同一类任务时，可以使用工作队列模式。

RabbitMQ会默认将消息按照顺序逐条分发给每个消费者，消费者可以手动设置ack，告知RabbitMQ当前消息已经被处理完成，RabbitMQ可以继续分发下一条消息。

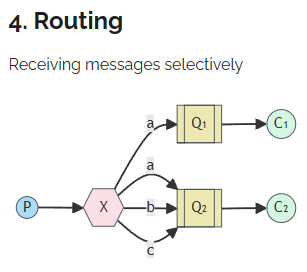
（3）发布/订阅模式



一个生产者、多个消费者，通过交换机（Exchange）实现一对多的消息传递。适用于需要将消息广播给多个消费者的情况。

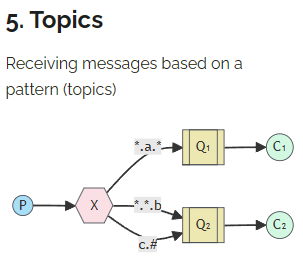
需要使用fanout类型的交换机，消息会无差别地发送给所有绑定到该交换机的队列。

（4）路由模式



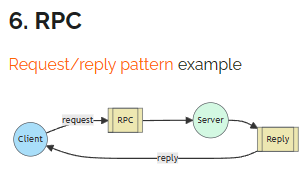
在发布/订阅模式的基础上，增加路由键（RoutingKey）的概念，实现消息的定向传递。生产者发送消息时指定路由键，交换机根据路由键将消息发送到匹配的队列中。

（5）主题模式



在路由模式的基础上，使用通配符来匹配路由键，实现更灵活的消息传递。通配符#匹配一个或多个词，\*匹配正好一个词。

（6）RPC模式



客户端发送请求消息到服务器，服务器处理请求并返回结果给客户端。实现远程调用功能，客户端和服务器通过RabbitMQ进行通信。需要定义请求队列和响应队列，以及处理请求的服务器和返回结果的服务器。

## QAMQP动态库

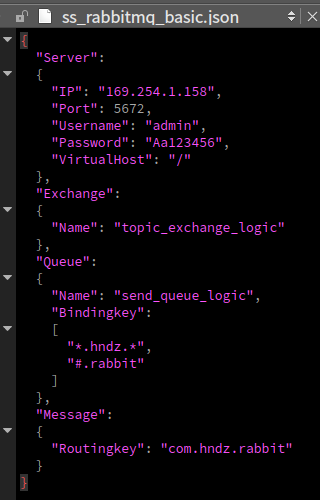
QAMQP动态库可用接口与交换机、队列设置可见下文件



## LibRabbitMQClient

LibRabbitMQClient是基于QAMQP动态库所设计的连接RabbitMQ服务器的工具库。

LibRabbitMQClient所使用的json文件格式如下：



LibRabbitMQClient的接口如下：



## 8. 总结

RabbitMQ 是一个功能强大、可靠性高的消息队列系统，通过 AMQP 协议提供了丰富的消息传递模式和灵活的配置选项，适用于各种异步通信和消息处理场景。在实际应用中，需要根据具体业务需求选择合适的交换机类型和消息传递模式，合理设计消息队列的结构和路由规则，以提升系统的性能、可靠性和扩展性。

## 补充

使用qamqp库时，建立socket连接与发送不建议在复杂场景下多线程使用，因为QAbstractSocket及其衍生类QSslSocket所创建的socket在Qt框架中并不直接支持多线程共享。在开发过程中原定方案为逻辑程序的主线程建立socket连接，在此之外建立一个发送消息的线程周期性向服务器发送数据，但在实际使用过程中会不断出现不同类型的frame error（帧错误）（unexpect frame, frame too large），虽然在设置等待帧写入时间上更改为不阻塞操作，但副作用为其不支持大数据量的发送数据。将发送线程合并至主线程，使用QTimer进行周期性的异步操作后，RabbitMQ客户端运行正常。