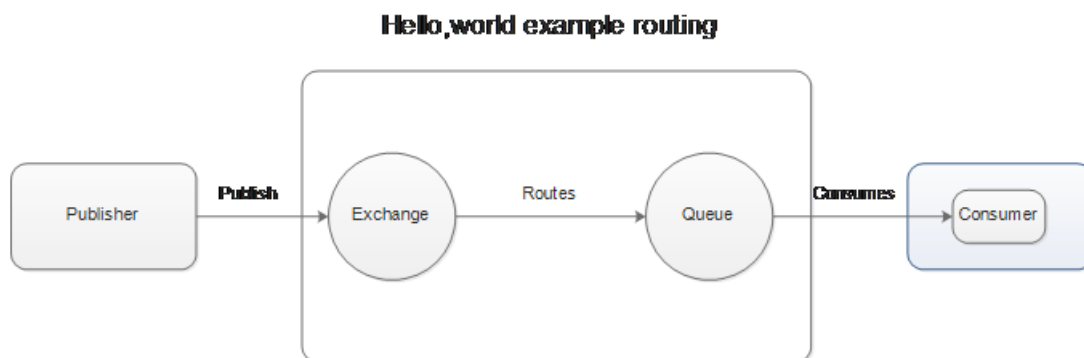


1.6 RabbitMQ 简介

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。2006年，AMQP 规范发布。类比HTTP。



rabbitMq中的重要概念

Broker

接收和分发消息的应用，RabbitMQ Server就是Message Broker

Virtual host

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

Connection

publisher / consumer 和 broker 之间的 TCP 连接

Channel

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。

Exchange

message 到达 broker 的第一站，根据分发规则，匹配查询表中的 routing key，分发消息到queue 中去。常用的类型有：direct (point-to-point, 直接投递, 精确匹配routing key), topic (publish-subscribe, 通配符匹配路由key) 和 fanout (multicast, 忽略routing key 直接发到所有队列中)

Queue

消息最终被送到这里等待 consumer 取走

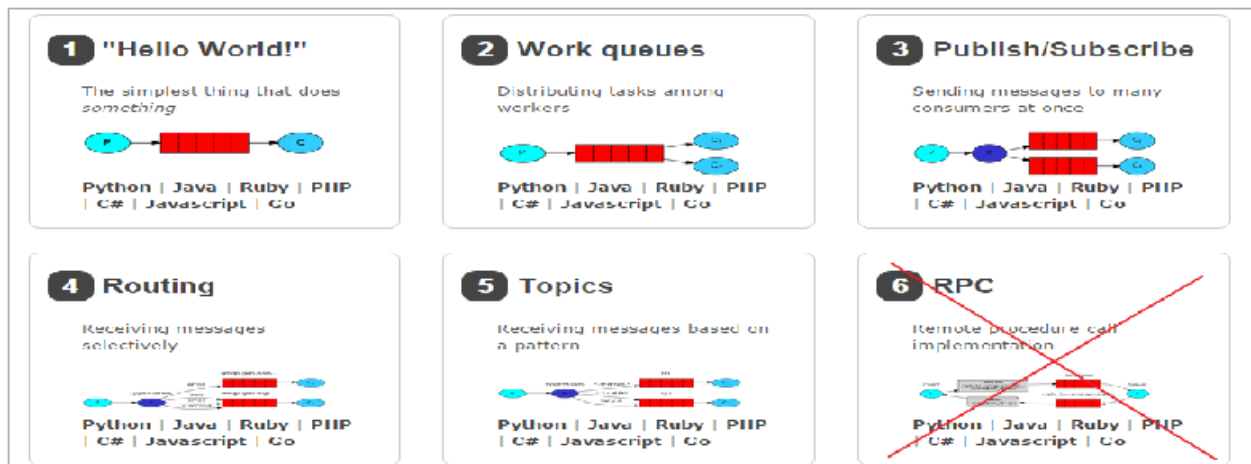
Binding

exchange 和 queue 之间的虚拟连接，binding 中可以包含 routing key。Binding信息被保存到exchange中的查询表中，用于message的分发依据

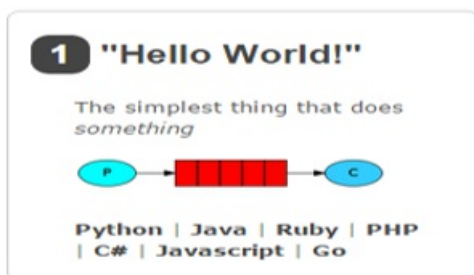
工作模式

1.6 RabbitMQ 简介

RabbitMQ 提供了 6 种工作模式：简单模式、work queues、Publish/Subscribe 发布与订阅模式、Routing 路由模式、Topics 主题模式、RPC 远程调用模式（远程调用，不太算 MQ；暂不作介绍）。
官网对应模式介绍：<https://www.rabbitmq.com/getstarted.html>

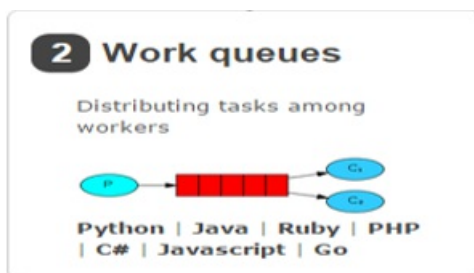


简单模式



生产者把消息直接发送到队列中去，消费者直接从队列中拉取消息

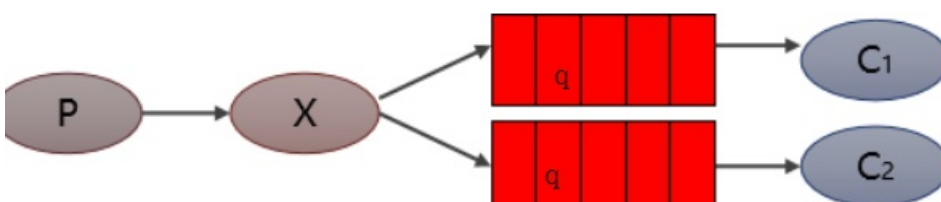
Work Queue



生产者把消息直接发送到队列中去，多个消费者直接从队列中拉取消息，Amqp协议内置了负载均衡策略，消息会平均分给所有消费者

Work Queues对于任务过重或任务较多情况使用工作队列可以提高任务处理的速度。

Pub/Sub 订阅模式



P:生产者(product),也就是要发送消息的程序，但是不再发送到队列中，而是发给X（交换机）

X:交换机(Exchange),用来处理消息路由,转发给队列,知道如何处理消息,例如递交给某个特别队列、递交给所有队列、或是将消息丢弃。到底如何操作,取决于Exchange的类型。Exchange有常见以下3种类型:

Fanout: 广播, 将消息交给所有绑定到交换机的队列

Direct: 定向, 把消息交给符合指定routing key 的队列

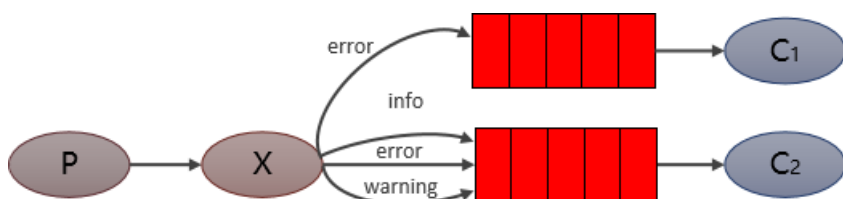
Topic: 通配符, 把消息交给符合routing pattern (路由模式) 的队列

q:队列(queue),消息队列, 接收消息、缓存消息

Cx:消费者, 消息的接收者, 会一直等待消息到来

生产者把消息发送到指定交换机(广播模式Fanout),交换机把消息分发给多个队列,消费者从队列中获取消息

Routing 路由模式



队列与交换机的绑定, 不能是任意绑定了, 而是要指定一个 RoutingKey (路由key)

消息的发送方在向 Exchange 发送消息时, 也必须指定消息的 RoutingKey

Exchange 不再把消息交给每一个绑定的队列, 而是根据消息的 Routing Key 进行判断, 只有队列的Routingkey与消息的Routing key 完全一致, 才会接收到消息

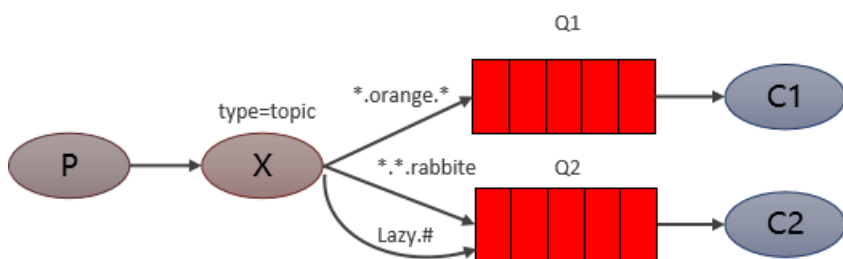
P: 生产者, 向 Exchange 发送消息, 发送消息时, 会指定一个routing key

X: Exchange (交换机), 接收生产者的消息, 然后把消息递交给与 routing key 完全匹配的队列

C1: 消费者, 其所在队列指定了需要 routing key 为 error 的消息

C2: 消费者, 其所在队列指定了需要 routing key 为 info、error、warning 的消息

Topics 通配符模式



Topic类型与Direct相比, 都是可以根据RoutingKey把消息路由到不同的队列。只不过Topic类型Exchange可以让队列在绑定Routing key的时候使用通配符!

Routingkey一般都是有一个或多个单词组成, 多个单词之间以"."分割, 例如: item.insert

通配符规则: #匹配一个或多个词, *匹配不多不少恰好1个词, 例如: item.# 能够匹配item.insert.abc或者item.insert, item.*只能匹配 item.insert

红色 Queue: 绑定的是 usa.#, 因此凡是以usa. 开头的Routingkey都会被匹配到。

黄色 Queue: 绑定的是 #.news, 因此凡是以.news 结尾的Routingkey都会被匹配。

工作模式总结

简单模式

一个生产者、一个消费者, 不需要设置交换机 (使用默认的交换机)。

工作队列模式Work Queue

一个生产者、多个消费者（竞争关系），不需要设置交换机（使用默认的交换机）。

发布订阅模式Publish/Subscribe

需要设置类型为fanout的交换机，并且交换机和队列进行绑定，当发送消息到交换机后，交换机会将消息发送到绑定的队列。

路由模式Routing

需要设置类型为direct的交换机，交换机和队列进行绑定，并且指定routing key，当发送消息到交换机后，交换机会根据routing key将消息发送到对应的队列。

通配符模式Topic

需要设置类型为topic的交换机，交换机和队列进行绑定，并且指定通配符方式的routing key，当发送消息到交换机后，交换机会根据routing key将消息发送到对应的队列。