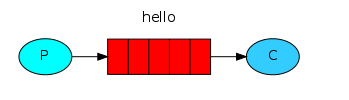
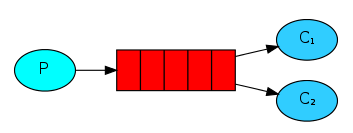
# 工作模式

## 简单模式



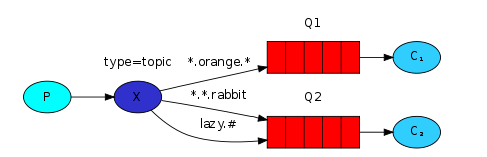
1. 消息产生着§将消息放入队列
2. 消息的消费者(consumer) 监听(while) 消息队列,如果队列中有消息,就消费掉,消息被拿走后,自动从队列中删除(隐患 消息可能没有被消费者正确处理,已经从队列中消失了,造成消息的丢失)应用场景:聊天(中间有一个过度的服务器;p端,c端)

## work工作模式（资源竞争）



1. 消息产生者将消息放入队列消费者可以有多个,消费者1,消费者2,同时监听同一个队列,消息被消费?C1 C2共同争抢当前的消息队列内容,谁先拿到谁负责消费消息(隐患,高并发情况下,默认会产生某一个消息被多个消费者共同使用,可以设置一个开关(syncronize,与同步锁的性能不一样) 保证一条消息只能被一个消费者使用)
2. 应用场景:红包;大项目中的资源调度(任务分配系统不需知道哪一个任务执行系统在空闲,直接将任务扔到消息队列中,空闲的系统自动争抢)

## Topic主题模式



1. 星号井号代表通配符
2. 星号代表多个单词,井号代表一个单词
3. 路由功能添加模糊匹配
4. 消息产生者产生消息,把消息交给交换机
5. 交换机根据key的规则模糊匹配到对应的队列,由队列的监听消费者接收消息消费

## Fanout Exchange形式

Fanout Exchange形式又叫广播形式。全部都可收到，也可叫做发布订阅