# 消息订阅发布简介

1. 消息订阅发布模式是一种常用的设计模式，它有3个角色：消息、订阅者、发布者。简单来讲，就是让多个订阅者同时监听某个发布者发布的消息，当这个消息发送成功，所有订阅者都会收到此消息，每条消息可以被多个订阅者订阅；订阅者订阅消息后才能接收到发布者发布的消息，在订阅前发布的消息，订阅者是接收不到的。这就是消息订阅发布模式。
2. Redis的消息订阅发布功能主要通过psubscribe、publish、pubsub、punsubscribe、subscribe、unsubscribe等命令来实现。一个客户端通过publish命令向订阅者发布消息时，这个客户端就是消息发布者；而当另一个客户端或多个客户端使用subscribe或psubscribe命令接收消息时，这个客户端就是消息订阅者。一个客户端可以通过subscribe命令同时订阅多个频道（Channel），当其他客户端（消息发布者）向被订阅的频道发布消息后，这个频道的消息订阅者都会接收到这个信息。
3. 频道（Channel）是消息发布者和消息订阅者的中介，当消息发布者将消息发布给频道，这个频道接收到消息后，负责把消息发送给这个频道的消息订阅者。

# 实现消息订阅发布

消息订阅发布是主要是通过publish、subscribe、psubscribe、unsubscribe、punsubscribe等命令来实现的。

## publish

publish命令用于发布消息给指定的频道。命令如下：

**publish channel message**

channel表示要发布到哪个频道；

message表示要发布的消息；

当客户端使用了publish命令向频道发布消息，该客户端就表示消息发布者。

返回值：返回一个整数，表示接收到这条消息的客户端数量。

## subscribe

subscribe命令用于订阅一个或多个频道。命令如下：

**subscribe channel [channel……]**

channel表示要订阅的频道，可以订阅多个；

客户端执行subscribe命令订阅频道后，进入订阅状态，将不能执行subscribe、psubscribe、unsubscribe、punsubscribe命令以外的其他命令，该客户端称为消息订阅者。

## psubscribe

psubscribe命令用于订阅符合给定匹配模式的频道。命令如下：

**psubscribe pattern [pattern…..]**

pattern表示匹配模式，可以由多个，当为\*时，表示订阅所有的频道；

·h?llo：？匹配一个字符，可以订阅hallo、hbllo等频道；

·h\*llo：\*匹配任意个字符，包括空字符，可以订阅h开头，以llo结尾的频道；

·h[ae]llo：[]表示多者之一，可以订阅hallo、hello频道；

如果遇到特殊字符，需要使用\符号进行转义；

返回值：返回的信息中包括了订阅成功频道的数量；

## unsubscribe

unsubscribe命令用于退订一个或多个频道。命令如下：

**unsubscribe [channel……]**

channel表示要退订的频道，可以有多个；当没有指定要退订的频道时，则表示退订所有频道。

注意：当客户端进入订阅状态后，是执行不了退订命令的，必须借助telnet之类的工具才能在订阅状态下执行退订操作。

## punsubscribe

punsubscribe命令用于退订符合给定匹配模式的频道。命令如下：

**punsubscribe [pattern……]**

pattern表示匹配模式；如果没有指定匹配模式或匹配模式指定为\*，则表示退订所有订阅的频道，但需要注意的是，punsubscribe只能退订通过psubscribe命令订阅的频道，它不会影响subscribe订阅的频道。

## pubsub

pubsum命令是一个自检命令，可用于检查发布/订阅子系统的状态。这个命令是由几个子命令组成的，命令如下：

**pubsub <subcommand> [argument [argument…]]**

子命令一：

pubsub channels命令用于返回已被订阅的频道。命令如下：

**pubsub channels [pattern]**

pattern表示匹配模式，如果没有指定，则返回所有已被订阅的频道；

返回值：以列表的形式返回已被订阅的频道。

子命令二：

pubsub numsub命令用于获取指定频道的订阅者数量。命令如下：

**pubsub numsub [channel……]**

channel表示要查看的订阅频道，可以给定多个；如果一个也没给定，则返回空列表。

返回值：以列表的形式返回给定频道的订阅者数量。

子命令三：

pubsub numpat命令用于获取所有客户端运行psubscribe命令的总次数。命令如下：

**pubsub numpat**

返回值：返回一个数值，表示使用所有客户端执行psubscribe命令的总次数。

# 消息订阅功能的订阅

## 订阅频道

订阅频道指的是使用subscribe命令订阅频道。

1. 当客户端使用subscribe命令订阅一个或多个频道后，这个客户端与被订阅的频道建立了一种订阅频道关系，这个关系将会保存到服务器状态的pubsub\_channels字典里面，这个字典也是一个键值对，键就是某个被订阅的频道，键的值是一个链表，这个链表中记录记录了所有订阅这个频道的客户端（消息订阅者）。
2. 当客户端执行unsubscribe命令来退订一个或多个频道时，服务器将从pubsub\_channels字典中删除被退订频道与客户端的关联关系。
3. 当客户端执行publish命令将消息发送给所有订阅这个频道的订阅者时，publish命令会在pubsub\_channels字典中查找所有订阅这个频道的订阅者名单（订阅者链表），然后将消息发送给名单中的所有订阅者。

订阅频道推送的消息有以下几种：

1. subscribe信息

subscribe信息表示客户端已经成功地订阅了指定的频道。在返回信息中：

第一个值是subscribe字符串，表示指定的频道订阅成功；

第二个值是订阅成的频道名称；

第三个值是目前已经成功订阅频道的数量。

1. unsubscribe信息

unsubscribe信息表示客户端已经成功取消订阅指定的频道。在返回信息中：

第一个值是unsubscribe字符串，表示已成功退订指定的频道；

第二个值是要退订的频道名称；

第三个值是当前客户端订阅的频道数量，如果为0，则表示这个客户端没有订阅任何频道。

1. message信息

message信息表示订阅频道的客户端已成功地收到了另一个客户端向这个频道发送过来的信息；换句话说，就是信息订阅者成功收到了消息发布者向频道发送的信息。

在返回信息中：

第一个值是message字符串，表示返回值的类型是消息；

第二个值是发送消息的频道名称；

第三个值是消息的内容。

## 订阅模式

订阅模式指的是使用psubscribe命令根据给定的匹配模式来订阅频道。

1. 当客户端执行psubscribe命令订阅某个或某个消息模式时，这个客户端与被订阅的消息模式之间就建立了一种订阅模式关系，这个订阅模式关系会被保存到服务器状态的pubsub\_patterns属性里面。pubsub\_patterns属性是一个链表，这个链表中的每个节点都包含着一个pubsubPattern结构体，这个结构体具有client和pattern两个属性，client属性表示订阅模式的客户端，而pattern属性表示被订阅的模式。
2. 当客户端执行punsubscribe命令来退订某个或某些订阅模式时，服务器将会从pubsub\_patterns链表中，根据punsubscribe命令指定的模式来查找与这个模式相匹配的订阅模式，然后删除pattern属性为退订模式，并且client属性为执行退订模式的客户端的pubsubPattern结构体。

根据模式订阅频道推送的消息格式有以下几种：

1、psubscribe信息

psubscribe信息表示客户端已经成功订阅了指定的模式。在返回信息中：

第一个值是psubscribe字符串，表示成功订阅；

第二个值是订阅的模式名称；

第三个值是当前已经订阅的模式数量。

2、punsubscribe信息

punsubscribe信息表示客户端已经成功退订了指定的消息模式。在返回的信息中：

第一个值是punsubscribe字符串，表示成功退订了指定的模式；

第二个值是退订的模式名称；

第三个值是客户端当前已经订阅的模式数量，如果为0，则表示这个客户端当前没有订阅任何模式。

3、pmessage信息

pmessage表示订阅模式的客户端已经成功地收到了另一个客户端向这个模式所对应的频道发送的消息。换句话说，就是消息订阅者成功收到了消息发布者所发送的信息。

在返回的信息中：

第一个值是pmessage字符串，表示成功接收到了信息；

第二个值是指定的模式名中的任意一个；

第三个值是发送消息的频道名；

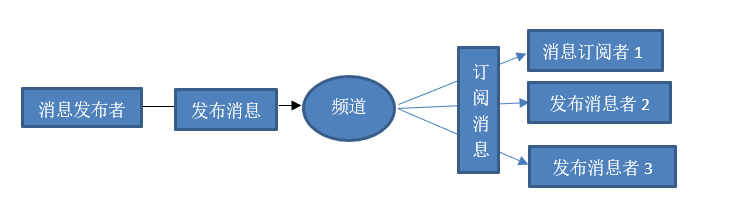
第四个值是消息内容。

# Redis消息队列

常见的消息队列有两种模式，分别是：消息订阅发布模式、消息生产者/消费者模式。Redis的消息队列也是基于这两种模式实现的。

## 消息订阅发布模式

消息发布者将消息发布到相应的频道（消息队列），消息订阅者通过订阅这个消息频道，就能接收到消息发布者所发送的消息。只要是订阅了这个消息频道的订阅者，它们接收到的信息都是一样的，也就是消息发布者将消息发送给了每位订阅者。



订阅消息

消息订阅者1

发布消息者2

发布消息

消息发布者

发布消息者3

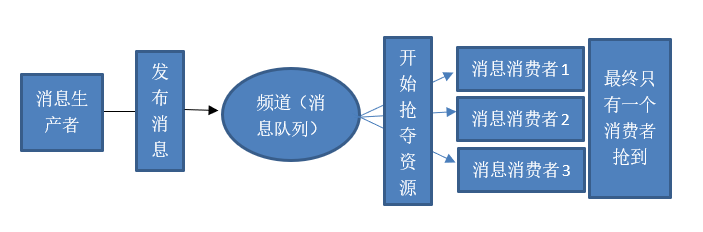
说明：当消息发布者发送一条信息到Redis Server后，只要消息订阅者订阅了该频道，就可以接受到该消息。同时，消息订阅者可以订阅不同的频道；

消息订阅发布模式的适用场景：微信公众号、订阅号的推送、新闻APP的推送，商城系统的信息推送等。

## 消息生产者/消费者模式

消息生产者将生产的消息放入消息队列中，多个消息消费者同时监听这个消息队列，谁先抢到消息，谁就从消息队列中取走消息去处理，即对于每条消息，只能被最多一个消费者消费。

与消息发布模式不同的是，消息发布模式不需要抢夺信息，每个订阅者都可以得到相同的信息；而消息生产者/消费者一种“抢”的模式，消息生产者没生产并发送一条消息，只能有一个消费者可以获取消息，谁的网速快、人品好，谁就可以获得那条消息。



说明：消息生产者/消费者模式也有多个消费者来消费，但是只能有一个消费者可以获得信息，其他消费者就只能继续监听消息队列，等待下一次抢信息。

消息生产者/消费者模式的适用场景：用户系统登录注册接收验证码、登录码，订单系统下单成功的短信，抢红包等。