**[RabbitMq 队列的一些常见方法及参数](https://www.cnblogs.com/fanqisoft/p/10389044.html)**

方法：

　　1、QueueDeclare　　声明队列

1 public static QueueDeclareOk QueueDeclare(String queue, Boolean durable, Boolean exclusive, Boolean autoDelete, IDictionary<String, Object> arguments);

queue:声明的队列名称

durable：是否持久化，是否将队列持久化到mnesia数据库中，有专门的表保存我们的队列声明。

exclusive：排外，①当前定义的队列是connection的channel是共享的，其他的connection是访问不到的。②当connection关闭的时候，队列将被删除。

autoDelete：自动删除，当最后一个consumer（消费者）断开之后，队列将自动删除。

arguments：参数是rabbitmq的一个扩展，功能非常强大，基本是AMPQ中没有的。  
  
2、QueueDeclareNoWait　　声明队列无需等待

3、QueueDeclarePassive　　消极的队列声明

　　被用来检测队列是否存在，不存在则抛出异常，存在则不做任何操作。

参数：

　　1.x-message-ttl：Number

　　1个发布的消息在队列中存在多长时间后被取消（单位毫秒）

　　\*可以对单个消息设置过期时间

　　2.x-expires：Number

　　当Queue（队列）在指定的时间未被访问，则队列将被自动删除。

　　3.x-max-length：Number

　　队列所能容下消息的最大长度。当超出长度后，新消息将会覆盖最前面的消息，类似于Redis的LRU算法。

　　4.x-max-length-bytes：Number

　　限定队列的最大占用空间，当超出后也使用类似于Redis的LRU算法。

　　5.x-overflow：String

　　设置队列溢出行为。这决定了当达到队列的最大长度时，消息会发生什么。有效值为Drop Head或Reject Publish。

　　6.x-dead-letter-exchange：String

　　有时候我们希望当队列的消息达到上限后溢出的消息不会被删除掉，而是走到另一个队列中保存起来。

　　7.x-dead-letter-routing-key：String

　　如果不定义，则默认为溢出队列的routing-key，因此，一般和6一起定义。

　　8.x-max-priority：Number

　　如果将一个队列加上优先级参数，那么该队列为优先级队列。

　　　　1）、给队列加上优先级参数使其成为优先级队列

　　　　x-max-priority=10【值不要太大，本质是一个树结构】

　　　　2）、给消息加上优先级属性

[复制代码](javascript:void(0);)

　　通过优先级特性，将一个队列编程stack（堆栈）

　　9.x-queue-mode：String

　　队列类型　　x-queue-mode=lazy　　懒队列，在磁盘上尽可能多地保留消息以减少RAM使用；如果未设置，则队列将保留内存缓存以尽可能快地传递消息。

　　10.x-queue-master-locator：String

　　将队列设置为主位置模式，确定在节点集群上声明时队列主位置所依据的规则。