

RocketMQ的几个重要概念

RocketMQ的前世今生

和大部分组件产生的原因类似，阿里巴巴内部为了适应淘宝 B2C 的更快、更复杂的业务，2001年启动了“五彩石项目”，阿里巴巴的第一代消息队列服务Notify就是在这个背景下产生的。

2010 年，阿里巴巴内部的 Apache ActiveMQ 仍然作为核心技术被广泛用于各个业务线，而顺序消息、海量消息堆积、完全自主控制消息队列服务，也是阿里巴巴同时期急需的。在这种背景下，2011年，MetaQ诞生。

2011年，LinkedIn将Kafka开源。2012年，阿里巴巴参考Kafka的设计，基于对MetaQ的理解和实际使用，研发了一套通用消息队列引擎，也就是 RocketMQ。自此才有了第一代真正的RocketMQ，2016年阿里云上线云RocketMQ消息队列服务。

2016年11月，阿里巴巴将RocketMQ捐献给Apache基金会。

Apache社区有一个很重要的理念：社区大于代码。虽然RocketMQ已经开源3年，在国内小有名气，而且在阿里巴巴被广泛应用并有较好的效果，但是依然不能达到 Apache优秀项目的标准。

在RocketMQ被捐献后，通过一系列的修改、评审、调整，悄悄升级至4.0版本，正式进入孵化阶段。

2017年09月25日，RocketMQ成功“毕业”（Apache社区项目孵化成功即为毕业），成为Apache 顶级项目，它是国内首个互联网中间件在 Apache 的顶级项目，也是继ActiveMQ、Kafka后Apache家族中全新一代消息队列引擎。

Apache RocketMQ

Apache RocketMQ™ is a unified messaging engine, lightweight data processing platform.

[Latest release v4.8.0](#)



13,424



7,335

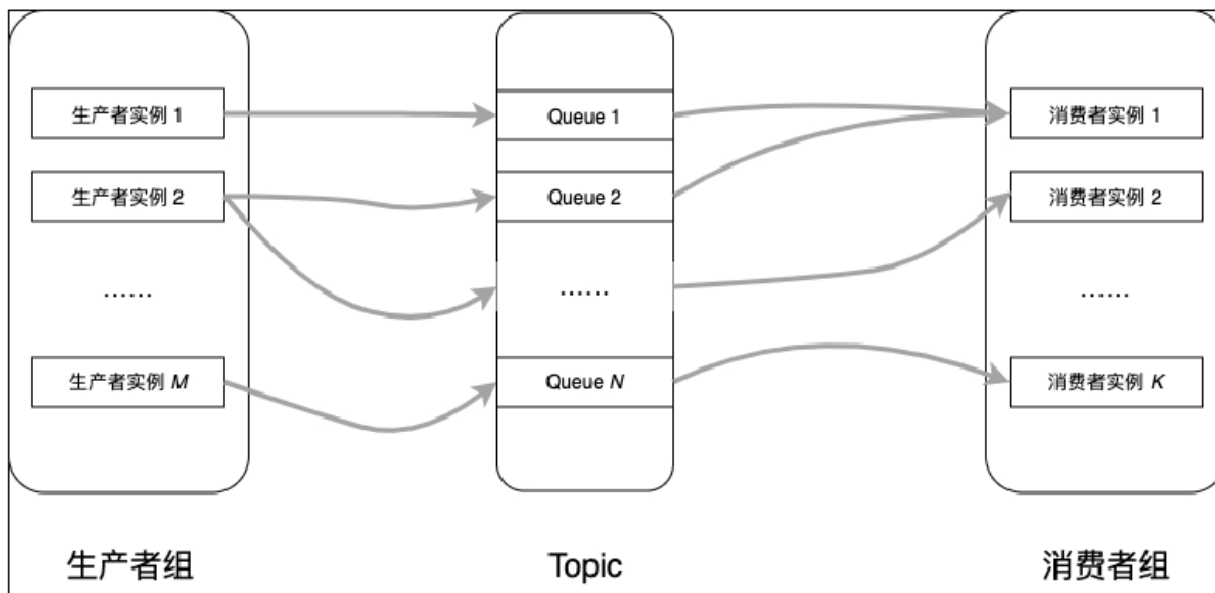
[Getting Started](#)



RocketMQ有很多优秀的特性，在可用性方面，RocketMQ强调集群无单点，任意一点高可用，客户端具备负载均衡能力，可以轻松实现水平扩容；在性能方面，在天猫双11大促背后的亿级消息处理就是通过RocketMQ提供的保障；在API方面，提供了丰富的功能，可以实现异步消息、同步消息、顺序消息、事务消息等丰富的功能，能满足大多数应用场景；在可靠性方面，提供了消息持久化、失败重试机制、消息查询追溯的功能，进一步为可靠性提供保障。

几个非常重要的概念

了解Rocket的诸多特性后，咱们来理解RocketMQ几个重要的概念：



- 生产者组Producer Group：一个逻辑概念，在使用生产者实例的时候需要指定一个组名。一个生产者组可以生产多个Topic的消息。
- 生产者实例Producer：一个生产者组部署了多个进程，每个进程都可以称为一个生产者实例。

- 主题Topic: 一个Topic代表消息的大类, 比如 “财政数据” Topic、 “商城订单” Topic, 一个Topic下可以拥有多个队列Queue
- 标签Tag: 二级分类, 对topic的进一步细化,用于区分同一个主题下的不同业务的消息, 例如: “财政数据” 的Tag可以有 “计划” 、 “指标” 、 “支付” , “商城订单” 的Tag可以有 “线上订单” 、 “线下订单” ...。
- 消息Message: 具体的数据消息, 每个message必须指定Topic
- 队列Queue: Queue就是消息队列的物理通信通道, 消息采用FIFO (先进先出) 模式传输, 负责向消费者Push (推送) 数据, 或者由消费者直接Pull (拉取)
- 消费者组Consumer Group: 一个逻辑概念, 在使用消费者时需要指定一个组名。一个消费者组可以订阅多个Topic。
- 消费者实例Consumer: 一个消费者组程序部署了多个进程, 每个进程都可以称为一个消费者实例。