

你都工作两年半了，还不会 RabbitMQ？

作者：做梦都在改BUG

2023-01-31 · 湖南

本文字数：3453 字

阅读完需约 15 分钟

What is rabbitMQ ?

RabbitMQ 是一个由 Erlang 语言开发的 **AMQP**(高级消息队列协议) 的开源实现。

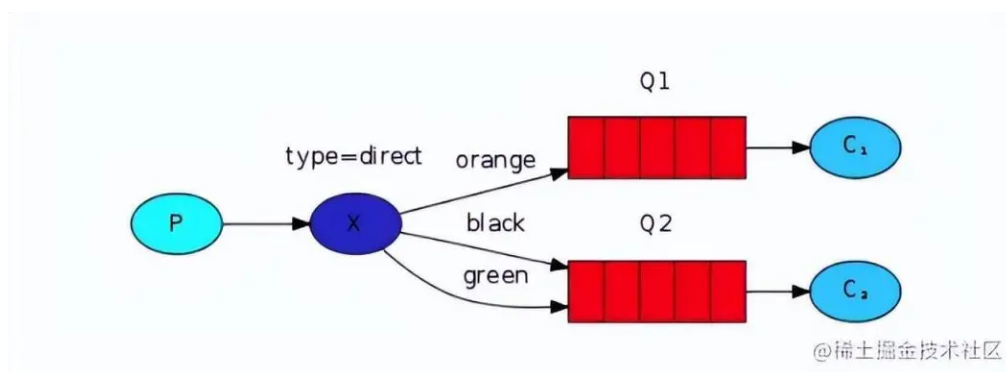
RabbitMQ 是**轻量级且易于部署**的，能支持多种消息协议。

RabbitMQ 可以部署在分布式和联合配置中，以满足**高规模、高可用性**的需求。

具体特点包括：

- **可靠性** (Reliability)：RabbitMQ 使用一些机制来保证可靠性，如持久化、传输确认、发布 确认。
- **灵活的路由** (Flexible Routing)：在消息进入队列之前，通过 Exchange 来路由消息的。对于典型的路由功能，RabbitMQ 已经提供了一些内置的 Exchange 来实现。针对更复杂的路由功能，可以将多个 Exchange 绑定在一起，也通过插件机制实现自己的 Exchange。
- **消息集群** (Clustering)：多个 RabbitMQ 服务器可以组成一个集群，形成一个逻辑 Broker。
- **高可用** (Highly Available Queues)：队列可以在集群中的机器上进行镜像，使得在部分节点出现问题的情况下队列仍然可用。
- **多种协议** (Multi-protocol)：RabbitMQ 支持多种消息队列协议，比如 STOMP、MQTT 等等。
- **多语言客户端** (Many Clients)：RabbitMQ 几乎支持所有常用语言，比如 Java、.NET、Ruby 等等。
- **管理界面** (Management UI)：RabbitMQ 提供了一个易用的用户界面，使得用户可以监控和管理消息 Broker 的许多方面。
- **跟踪机制** (Tracing)：如果消息异常，RabbitMQ 提供了消息跟踪机制，使用者可以找出发生了什么。
- **插件机制** (Plugin System)：RabbitMQ 提供了许多插件，来从多方面进行扩展，也可以编写自己的插件。

RabbitMQ 的消息模型



标志	中文名	英文名	描述
P	生产者	Producer	消息的发送者，可以将消息发送到交换机
C	消费者	Consumer	消息的接收者，从队列中获取消息进行消费
X	交换机	Exchange	接收生产者发送的消息，并根据路由键发送给指定队列
Q	队列	Queue	存储从交换机发来的消息
type	交换机类型	type	direct表示直接根据路由键（orange/black）发送消息

@稀土掘金技术社区

Why use rabbitMQ ?

下面，我以（花店）商家的角色向大家形象地举例：

异步

之前顾客来店里下单，我会叫顾客等一下，同时叫店员准备订单，准备好送到顾客手上了顾客**才能离开**

现在顾客打电话给我：“我要买 xxx，地址是：xxx，你帮我送一下”

我拿个小本子记下：顾客 a，电话：xxx，地址：xxx

店员**有空**后就会准备订单并配送

解耦

以前有新订单时，我会亲自找**每一个店员**（负责准备花束的，负责记账的，负责送花的等），告诉他们有新订单了，有空了处理一下

如果有店员入职，我通知的时候会多**通知**一个人；离职时，少**通知**一个人（维护一个需要通知的人员列表）

现在，有新订单的时候，我只需要**记到小本子上**，店员有空了自己来看

削峰

去年七夕，很多电话打给我，我把每一个订单告诉店员，但是店员忙不过来，客户又一直打电话来催，最后店员累成狗直接罢工

今年七夕节我学乖了，电话打进来我会告诉顾客：“我知道了，**会尽快安排处理**”，然后记到小本子上就行，店员有空时**按顺序来处理订单**就好

另外还有一种思路，引导客户不一定要在七夕才开始下单，可以**提前先买**（淘宝的双十一预售就是出于这样的削峰思路）

以上是 rabbitMQ 解决的**核心问题**。

How to use rabbitMQ ?

基操安装方式

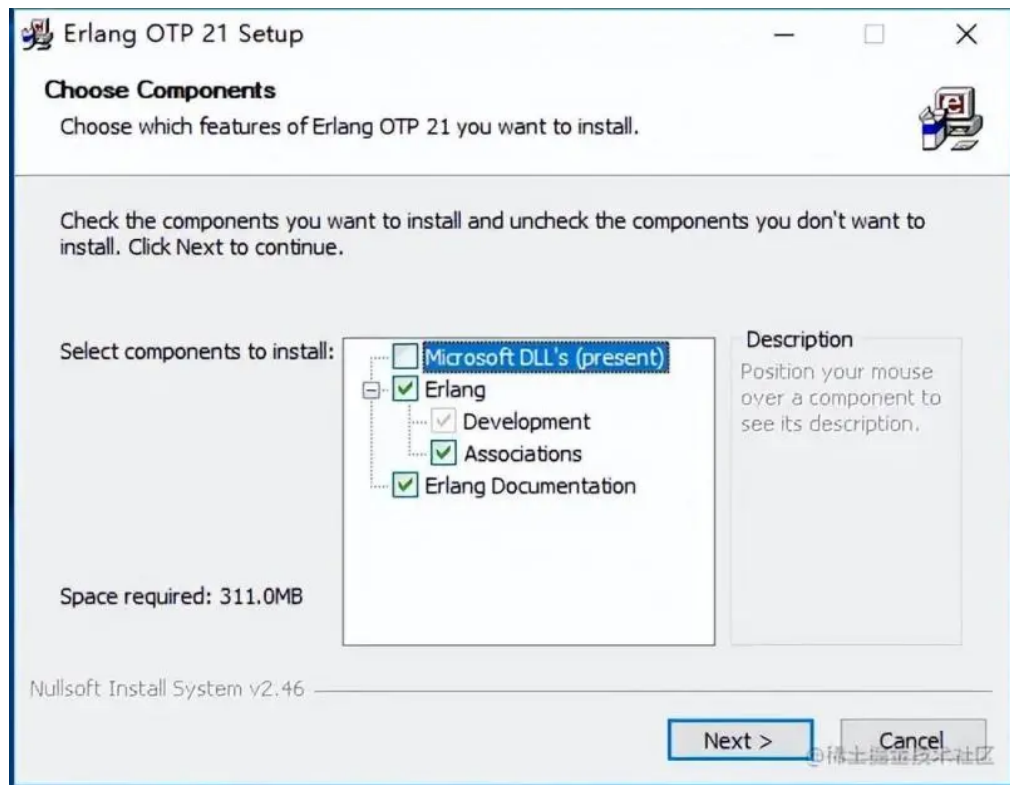
MAC 端

```
1 brew install rabbitmq
```

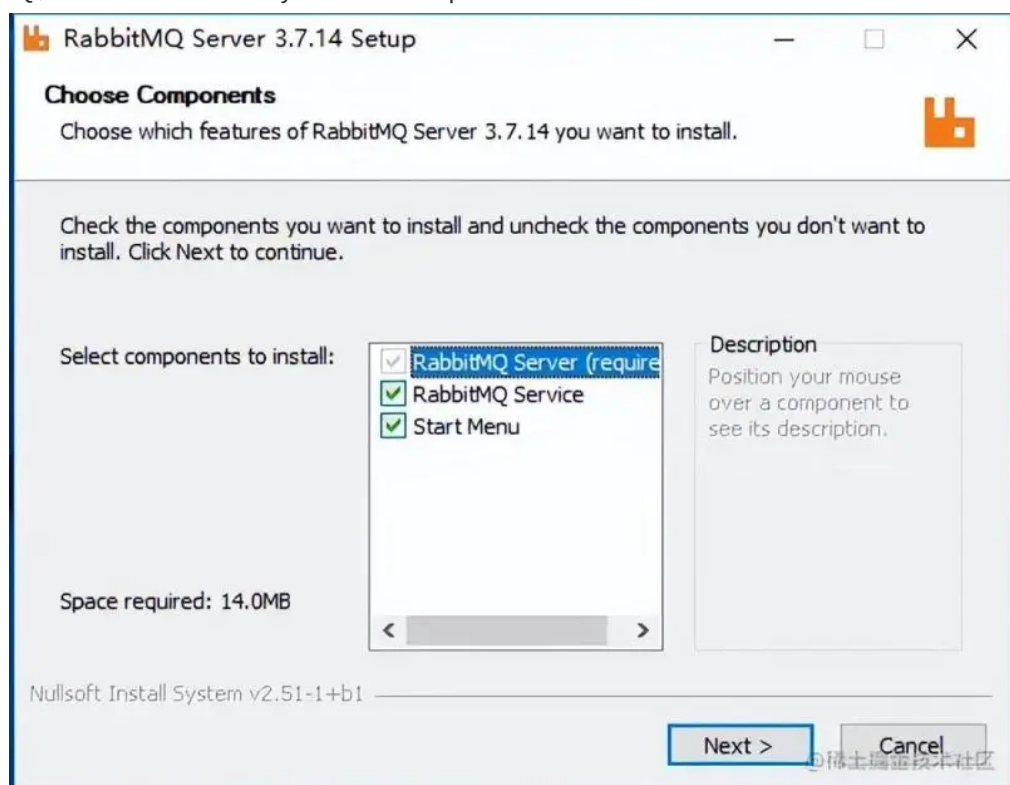
[复制代码](#)

Windows 端

1. 安装 Erlang，下载地址：[erlang.org/download/ot...](http://erlang.org/download/otp...)



2. 安装 RabbitMQ，下载地址：dl.bintray.com/rabbitmq/al...



3.安装完成后，进入 RabbitMQ 安装目录下的 sbin 目录

台电脑 > 本地磁盘 (D:) > developer > env > RabbitMQ Server > rabbitmq_server-3.7.14 > sbin

名称	修改日期	类型	大小
cuttlefish	2019/3/29 5:47	文件	458 KB
rabbitmqctl.bat	2019/3/29 5:47	Windows 批处理...	3 KB
rabbitmq-defaults.bat	2019/3/29 5:47	Windows 批处理...	2 KB
rabbitmq-diagnostics.bat	2019/3/29 5:47	Windows 批处理...	3 KB
rabbitmq-echo.bat	2019/3/29 5:47	Windows 批处理...	2 KB
rabbitmq-env.bat	2019/3/29 5:47	Windows 批处理...	17 KB
rabbitmq-plugins.bat	2019/3/29 5:47	Windows 批处理...	3 KB
rabbitmq-server.bat	2019/3/29 5:47	Windows 批处理...	11 KB
rabbitmq-service.bat	2019/3/29 5:47	Windows 批处理...	14 KB

@稀土掘金技术社区

4.在地址栏输入 cmd 并回车启动命令行，然后输入以下命令启动管理功能：

```
1 rabbitmq-plugins enable rabbitmq_management
```

复制代码

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17134.765]
(c) 2018 Microsoft Corporation。保留所有权利。

D:\developer\env\RabbitMQ Server\rabbitmq_server-3.7.14\sbin>rabbitmq-plugins enable rabbitmq_management
Enabling plugins on node rabbit@DESKTOP-K1F707Q:
rabbitmq_management
({plugins_not_found, [:rabbitmq_delayed_message_exchange]})

D:\developer\env\RabbitMQ Server\rabbitmq_server-3.7.14\sbin>
```

@稀土掘金技术社区

5.访问地址查看是否安装成功：http://localhost:15672/



CentOS 端

安装 erlang

```
1 # rabbitmq依赖erlang 需要自己去自行下载
2 cd /path/to/erlang-sound-code && ./configure --prefix=/usr/local/erlang
3 make && make install
4
5 vim /etc/profile
6 # 添加
7 export PATH=$PATH:/usr/local/erlang/bin
8
9 source /etc/profile
10 # 输入erl, 会出现版本信息, 即安装成功
```

复制代码

安装 rabbitmq

复制代码

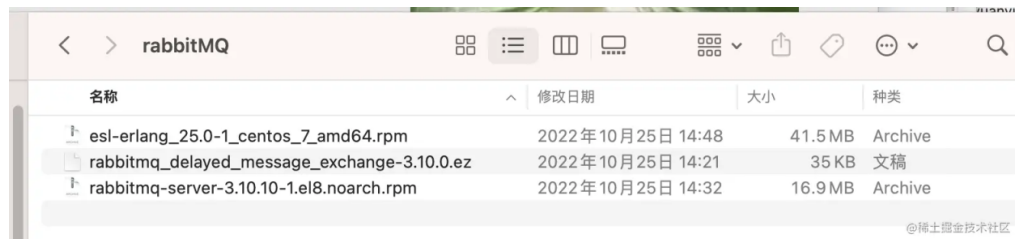
```
1 #下载 abbitmq_server-3.8.16 并移动到/usr/local/下
2 vim /etc/profile
3 # 添加
4 export PATH=$PATH:/usr/local/rabbitmq_server-3.8.16/sbin
5 source /etc/profile
6
7 cd /usr/local/rabbitmq_server-3.8.16/sbin
8 # 启动
9 ./rabbitmq-server start
```

功能实现

RabbitMQ 实现延迟消息的方式有两种，一种是使用死信队列实现，另一种是使用延迟插件实现。
死信队列的实现网上较多，本文介绍更简单的，使用延迟插件实现（mac 环境，java 版本）。

另外的安装方式（建议使用这种）

首先准备需要用到的安装文件及插件(rabbitmq_delayed_message_exchange)，版本需要匹配，不匹配的版本可能装不上或导致兼容问题。



本人使用的 erl_25.0 和 rabbitMQ-3.10.0（可以到官网下载或者私信作者获取）。使用这种方式安装的优点在于本地安装和服务器安装流程完全一致，不过服务器需要开放安全端口 5672,15672 视情况，一般建议测试环境开放，生产环境关闭。

安装 erl 和 rabbitMQ，具体步骤略（这个应该没人不会吧，逃~）。

将插件文件复制到 RabbitMQ 安装目录的 plugins 目录下，执行以下命令后重启 rabbitMQ：

```
1 rabbitmq-plugins enable rabbitmq_delayed_message_exchange
```

复制代码

实现延迟消息

以一个实际业务场景举例：当客服状态为在线且 3 分钟未回复客户消息时，自动重启 im 会话机器人接管会话。这是一个常见的延迟消息使用场景。

首先在 pom.xml 文件中添加 AMQP 相关依赖

```
1 <!-- 消息队列相关依赖 -->
2 <dependency>
3     <groupId>org.springframework.boot</groupId>
4     <artifactId>spring-boot-starter-amqp</artifactId>
5 </dependency>
```

复制代码

在 application.yml 添加 RabbitMQ 的相关配置

[复制代码](#)

```
1 spring:
2   rabbitmq:
3     host: localhost # rabbitmq的连接地址
4     port: 5672 # rabbitmq的连接端口号
5     virtual-host: /mall # rabbitmq的虚拟host
6     username: im # rabbitmq的用户名
7     password: xxxxxx # rabbitmq的密码
8     publisher-confirms: true #如果对异步消息需要回调必须设置为true
```

接下来创建 RabbitMQ 的 java 配置，主要用于配置交换机、队列和绑定关系

[复制代码](#)

```
1 /**
2  * 消息队列配置
3  */
4 @Configuration
5 public class RabbitMqConfig {
6   /**
7    * 机器人消息重启插件消息队列所绑定的交换机
8    */
9   @Bean
10  CustomExchange chatPluginDirect() {
11    //创建一个自定义交换机，可以发送延迟消息
12    Map<String, Object> args = new HashMap<>();
13    args.put("x-delayed-type", "direct");
14    return new CustomExchange(QueueEnum.QUEUE_RESET_MESSAGE_CANCEL.getExchange(), "x-delayed-me
15  }
16
17  /**
18   * 机器人消息重启插件队列
19   */
20  @Bean
21  public Queue chatPluginQueue() {
22    return new Queue(QueueEnum.QUEUE_RESET_MESSAGE_CANCEL.getName());
23  }
24
25  /**
26   * 将机器人消息重启插件队列绑定到交换机
27   */
28  @Bean
29  public Binding chatPluginBinding(CustomExchange chatPluginDirect, Queue chatPluginQueue) {
30    return BindingBuilder
31      .bind(chatPluginQueue)
32      .to(chatPluginDirect)
33      .with(QueueEnum.QUEUE_RESET_MESSAGE_CANCEL.getRouteKey())
34      .noargs();
35  }
36 }
```

创建一个消息的发出者,通过给消息设置 x-delay 头来设置消息从交换机发送到队列的延迟时间

[复制代码](#)

```
1 /**
2  * 机器人重启队列发出者
3  */
4 @Component
5 @Slf4j
6 public class ChatQueueSender {
7   private static Logger LOGGER = LoggerFactory.getLogger(ChatQueueSender.class);
```

```

8      @Autowired
9      private AmqpTemplate amqpTemplate;
10
11     public void sendMessageToChat(Long cmid, final long delayTimes) {
12         //给延迟队列发送消息
13         amqpTemplate.convertAndSend(QueueEnum.QUEUE_RESET_MESSAGE_CANCEL.getExchange(), QueueEnum.Q
14             @Override
15             public Message postProcessMessage(Message message) throws AmqpException {
16                 //给消息设置延迟毫秒值
17                 message.getMessageProperties().setHeader("x-delay", delayTimes);
18                 return message;
19             }
20         });
21     }
22 }

```

创建一个消息的接收者，用于处理延迟插件队列中的消息。

```

1  /**
2   * 机器人重启队列处理器
3   */
4  @Component
5  @Slf4j
6  @RabbitListener(queues = "im.chat.cancel")
7  public class ChatQueueReceiver {
8      @Autowired
9      private ChatRestartRobotService chatRestartRobotService;
10
11      @RabbitHandler
12      public void handleOnChat(Long cmid) {
13          // log.info("机器人会话重启");
14          chatRestartRobotService.restartRobot(cmid);
15      }
16  }

```

复制代码

最后，在对应的地方调用即可：

```

// 存在该条消息记录 顾问在线 机器人未禁用
if (selectChat.getStatus() == 2 && selectChat.getIsDisableRobot() == 2) {
    // 顾问在线 启动延时任务
    chatRestartRobotService.addRestartRobotQueue(cmid);
    return CommonResult.success(data: null, message: "接收成功");
}

```

@稀土掘金技术社区

作者：断律绎殇

链接：<https://juejin.cn/post/7173571716294115336>

来源：稀土掘金

发布于：2023-01-31 | 阅读数：94

Java

RabbitMQ

消息队列

消息中间件

+ 关注



还未添加个人签名 · 2021-07-28 加入

公众号：该用户快成仙了

👍 点赞

★ 收藏

💬 微信

🐦 微博

🏠 部落

🚩 举报

评论

快抢沙发！虚位以待

发布

• 暂无评论 •



促进软件开发及相关领域知识与创新的传播

InfoQ

关于我们

我要投稿

合作伙伴

加入我们

关注我们

联系我们

内容投稿：editors@geekbang.com

业务合作：hezuo@geekbang.com

反馈投诉：feedback@geekbang.com

加入我们：zhaopin@geekbang.com

联系电话：010-64738142

地址：北京市朝阳区叶青大厦北园

InfoQ 近期会议

北京 ArchSummit全球架构大会

上海 ArchSummit全球架构大会

广州 QCon全球软件开发大会

