# 1 简介

## 消息中间件（消息队列）

什么叫消息队列

消息（Message）是指在应用间传送的数据。消息可以非常简单，比如只包含文本字符串，也可以更复杂，可能包含嵌入对象。

消息队列（Message Queue）是一种应用间的通信方式，消息发送后可以立即返回，由消息系统来确保消息的可靠传递。消息发布者只管把消息发布到 MQ 中而不用管谁来取，消息使用者只管从 MQ 中取消息而不管是谁发布的。这样发布者和使用者都不用知道对方的存在。

消息中间件在互联网公司的使用中越来越多，消息中间件最主要的作用是解耦，中间件最标准的用法是生产者生产消息传送到队列，消费者从队列中拿取消息并处理，生产者不用关心是谁来消费，消费者不用关心谁在生产消息，从而达到解耦的目的。在分布式的系统中，消息队列也会被用在很多其它的方面，比如：分布式事务的支持，RPC的调用等等。

## 1.2 常用消息中间件

ActiveMQ 、RabbitMQ、阿里巴巴自主开发RocketMQ（捐给了apache）、亚马逊的SMS、ZeroMQ、炙手可热的Kafka

## 1.3 RabbitMQ介绍

RabbitMQ 即一个消息队列，主要是用来实现应用程序的异步和解耦，同时也能起到消息缓冲，消息分发的作用。

RabbitMQ是实现AMQP（高级消息队列协议）的消息中间件的一种，最初起源于金融系统，用于在分布式系统中存储转发消息，在易用性、扩展性、高可用性等方面表现不俗。RabbitMQ主要是为了实现系统之间的双向解耦而实现的。当生产者大量产生数据时，消费者无法快速消费，那么需要一个中间层。保存这个数据。

AMQP，即Advanced Message Queuing Protocol，高级消息队列协议，是应用层协议的一个开放标准，为面向消息的中间件设计。消息中间件主要用于组件之间的解耦，消息的发送者无需知道消息使用者的存在，反之亦然。AMQP的主要特征是面向消息、队列、路由（包括点对点和发布/订阅）、可靠性、安全。

RabbitMQ是一个开源的AMQP实现，服务器端用Erlang语言编写，支持多种客户端，如：Python、Ruby、.NET、Java、JMS、C、PHP、ActionScript、XMPP、STOMP等，支持AJAX。用于在分布式系统中存储转发消息，在易用性、扩展性、高可用性等方面表现不俗。

# 2 安装

## 2.1 下载erlang和rabbitmq-server的rpm

命令如下：

下载erlang

wget http://www.rabbitmq.com/releases/erlang/erlang-19.0.4-1.el7.centos.x86\_64.rpm

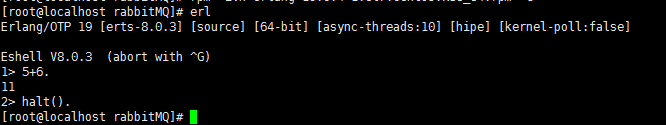
下载rabbitmq-server

wget <http://www.rabbitmq.com/releases/rabbitmq-server/v3.6.6/rabbitmq-server-3.6.6-1.el7.noarch.rpm>

## 2.2 安装erlang:

rpm -ivh erlang-19.0.4-1.el7.centos.x86\_64.rpm

检查erlang 是否安装成功



## 2.3 安装rabbitmq

rpm -ivh rabbitmq-server-3.6.6-1.el7.noarch.rpm

在安装rabbitmq时提示依赖socat

安装socat，如下命令：

　　　　 yum install socat

然后rpm -ivh rabbitmq-server-3.6.6-1.el7.noarch.rpm

# 3 常用管理命令学习

## 3.1启动和关闭

systemctl start rabbitmq-server # 启动

systemctl stop rabbitmq-server # 关闭

systemctl status rabbitmq-server # 查看状态

## 3.2 使用管理命令

cd /sbin

查看插件列表：./rabbitmq-plugins list

查看rabbitmq状态：./rabbitmqctl status

添加用户 ：./rabbitmqctl add\_user admin admin

设置用户角色：./rabbitmqctl set\_user\_tags admin administrator

列出所有用户：./rabbitmqctl list\_users

启用rabbitmq管理控制台：rabbitmq-plugins enable rabbitmq\_management

防火墙相关命令：

#centos7查看防火墙所有信息

firewall-cmd --list-all

#centos7查看防火墙开放的端口信息

firewall-cmd --list-ports

# 开放端口

firewall-cmd --zone=public --add-port=15672/tcp --permanent

#重启防火墙

systemctl restart rabbitmq-server

# 4 访问管理控制台

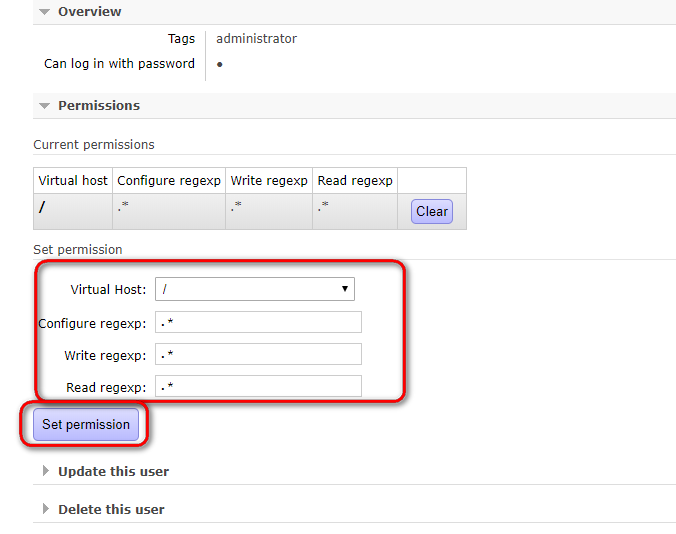
<http://10.6.51.230:15672/>

管理控制台访问默认端口15672

Admin授权



单击上图中admin用户，进入下图



单击上图中 Set permission

# 5 springboot 集成rabbitmq

5.1 pom文件中引入依赖

|  |
| --- |
| <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-amqp</artifactId>  </dependency> |

5.2 application.yml 中设置相关属性

|  |
| --- |
| spring.rabbitmq.host=192.168.0.86  spring.rabbitmq.port=5672  spring.rabbitmq.username=admin  spring.rabbitmq.password=123456 |

5.3 队列配置

|  |
| --- |
| @Configuration  public class RabbitConfig {  @Bean  public Queue Queue() {  return new Queue("hello");  }  } |

5.4 发送者

|  |
| --- |
| @Component  public class HelloSender {  // rabbitTemplate是springboot 提供的默认实现  @Autowired  private AmqpTemplate rabbitTemplate;  public void send() {  String context = "hello " + new Date();  System.out.println("Sender : " + context);  this.rabbitTemplate.convertAndSend("hello", context);  }  } |

5.5 接收者

|  |
| --- |
| @Component  @RabbitListener(queues = "hello")  public class HelloReceiver {  @RabbitHandler  public void process(String hello) {  System.out.println("Receiver : " + hello);  }  } |

5.6 测试

|  |
| --- |
| @RunWith(SpringRunner.class)  @SpringBootTest  public class RabbitMqHelloTest {  @Autowired  private HelloSender helloSender;  @Test  public void hello() throws Exception {  helloSender.send();  }  }  //注意，发送者和接收者的queue name必须一致，不然不能接收 |