**RabbitMQ**

看了篇博客，讲的是真的长啊，不过看到最后其实没有想象的难

介绍：

RabbitMQ是一个以amqp（高级消息队列协议（听听多高大上的协议名字））为底层实现的消息队列中间件。（知道他是erlang语言编译的就行了）

知识点：

虚拟主机、交换机、队列

路由key

四种交换机类型

Direct：direct 类型的行为是”先匹配, 再投送”. 即在绑定时设定一个 routing\_key, 消息的routing\_key 匹配时, 才会被交换器投送到绑定的队列中去.

特点：

按照一定的配置制度 根据路由键和队列的绑定键匹配去发送消息，如果有消息能匹配到则发送，否则丢弃，一个队列可以有多个路由键去匹配。

Topic：按规则转发消息（最灵活）

特点：和Direct类型类似都是按照路由键去匹配队列，但是不同的是Topic类型是根据通配符去匹配消息的,重点是他的匹配串必须带（.），\*.代表一个单词，#代表多个单词。最常用

Headers：设置 header attribute 参数类型的交换机

Fanout：转发消息到所有绑定队列

特点：

广播消息到与当前交换机绑定的所有队列上，忽略路由键。

今天0912 明天就是中秋节了，早上把上次没看完的看了一遍，总体来说，博客讲的还是不错的。

总结来说就是：rabbitMq下载安装（通常需要配置集群，避免宕机而丢失消息数据）、 配置较多（一种是原生的：

写一个配置类：

连接工厂、服务地址、端口、虚拟机名、用户名、密码、获取工程连接），

生产者发送消息到队列：

获取连接、创建通道、声明队列、发送消息

消费者：

获取连接、创建通道、声明队列、设置消息接收频率、定义队列的消费者、监听队列（可设置消息的回执为自动和手动）、while循环从运送回来的消息体中去取消息、给队列发送回执。

以上就是底层的消息使用，还有一些应用没有说，比如交换机的四种类型，也可以进行相应的配置，虽说配置不是很复杂，但是冗余。

Spring 就为这个中间件做了一个简单了封装

需要加一系列的配置才能够使用。

Springboot 为这个中间件做了一个很好的封装

只需要简单的配置

rabbitMq的地址、端口、登录名、密码

单独配置队列配置类

@Bean

public Queue queue() {

return new Queue("q\_hello");

}

多个队列依次往下添加就行，如果是主题模式，需要声明主题类型的交换机，并且将需要的队列与之绑定。

交换机的：

主题模式特点：

可以根据路由key绑定不同的队列，从而更加灵活

Fanout Exchange（订阅模式）（翻译为展开交换机）

相当于广播消息，让与之绑定的队列都接收到生产的消息

生产者发送消息this.rabbitTemplate.convertAndSend("q\_hello", context);

消费者只需加监听注解

@Component

@RabbitListener(queues = "q\_hello")

public class HelloReceiver2 {

@RabbitHandler

public void process(String hello) {

System.out.println("Receiver2 : " + hello);

}

}

多个消费者分别配置