消息队列方案

rabbit mq

# rabbit mq学习

http://rabbitmq-into-chinese.readthedocs.org/zh\_CN/latest/

# 项目中发送及接收mq消息

在numen、plus、admin不同的项目都有一个mq的配置文件，通常命名为applicationContext-rabbitmq.xml，在/src/main/resources/conf下

该文件中分三部分：公共配置（连接配置、消息转换bean、rabbitAdmin和rabbitTemplate）、测试mq实例配置、业务mq实例配置。详细看文件

**使用过程**

1. 声明队列bean，参考已有队列；
2. 确定消息对应的交互机（可以复用旧有的交互机，如默认的交换机e.default.${p.mq.flag}；或者声明一个新的交换机）。注意考虑新的消息类型是否广播还是主题类型，通常设置为直连类型交换机即可。
3. 交换机中进行绑定声明，将交换机和队列绑定起来，并指定路由key（交换机根据发送消息时指定的路由key会将消息发送到这个交换机下有这个路由key的所有队列）。若是广播交换机，不需要指定路由key；
4. 通过给交换机下的所有队列指定相同的路由key，也可以达到广播效果；
5. 通过将监听器bean和队列bean关联起来，为队列指定消费者。可以给同一个队列指定多个监听器，也可以一个监听器监听多个队列；

配置完成后，即可编写消息发送部分和消息接收部分的代码

1. 发送消息的对象类中注入

@Inject

**private** RabbitTemplate rabbitTemplate;

发送

rabbitTemplate.convertAndSend("exchangeName","routing-key", message);

1. 编写监听器类

@Component("corpDisbandMqListener")

**public** **class** CorpDisbandMqListener **extends** RabbitMQListener<Integer>{

@Override

**public** **void** handleMessage(Integer companyId) {

....

}

}

注意要继承RabbitMQListener或其子类FanoutRabbitMQListener、DirectRabbitMQListener；

父类中的模板类需要和handleMessage中的参数类型相同；

使用@Component注解

注意一些常量

mq.flag常量需要在每一个交换机和队列名称后面（因为多个测试环境公用了同一个mq服务器导致需要这么干，线上mq.flag可以是空的）

在Constants.MQ类中定义队列名、交换机名和routing-key

# 设计方案

主要是利用spring-rabbit的标签实现配置。

另外还有三个抽象类，用于监听器的编写。

RabbitMQListener、DirectRabbitMQListener、FanoutRabbitMQListener，一般情况下只需要继承RabbitMQListener，FanoutRabbitMQListener主要是为了能够创建随机队列（也就是每一个消费者都生成自己的队列，并绑定到同一个交换机）。DirectRabbitMQListener的作用是无法在配置文件中设置绑定关系，需要代码里设置，这个时候就需要继承这个父类。