## spring boot 整合rabbitmq

spring boot的环境怎么搭建这边就不提了， 这里引入spring boot -AMQP的依赖：

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

### 配置连接,创建交换机，队列

首先 我和上面一样 先要配置连接信息这里 可以选用yml的方式 也可以选用javaConfig的方式 这里两种方式我都贴出来 你们自己选

yml： 参数什么意思刚刚介绍过了 这里吧你自己的参数填进去就好了

spring:  
 rabbitmq:  
 host:  
 port:  
 username:  
 password:  
 virtual-host:

这样 spring boot 会帮你把rabbitmq其他相关的东西都自动装备好，

javaConfig：

@Bean  
public ConnectionFactory connectionFactory() {  
 CachingConnectionFactory connectionFactory = new CachingConnectionFactory("localhost",5672);  
 //我这里直接在构造方法传入了  
 // connectionFactory.setHost();  
 // connectionFactory.setPort();  
 connectionFactory.setUsername("admin");  
 connectionFactory.setPassword("admin");  
 connectionFactory.setVirtualHost("testhost");  
 //是否开启消息确认机制  
 //connectionFactory.setPublisherConfirms(true);  
 return connectionFactory;  
}

配置完连接之后 我们就可以开始发送消息和接收消息了（因为我们上面刚刚测试rabbitmq的时候创建过队列和交换机等等这种东西了 当然 spring boot也可以创建）

spring boot创建交换机 队列 并绑定：

@Bean  
public DirectExchange defaultExchange() {  
 return new DirectExchange("directExchange");  
}  
  
@Bean  
public Queue queue() {  
 //名字 是否持久化  
 return new Queue("testQueue", true);  
}  
  
@Bean  
public Binding binding() {  
 //绑定一个队列 to: 绑定到哪个交换机上面 with：绑定的路由建（routingKey）  
 return BindingBuilder.bind(queue()).to(defaultExchange()).with("direct.key");  
}

发送消息:

发送消息比较简单， spring 提供了一个RabbitTemplate 来帮助我们完成发送消息的操作

如果想对于RabbitTemplate 这个类进行一些配置（至于有哪些配置我们后面会讲到） 我们可以在config类中 吧他作为Bean new出来并配置

@Bean  
public RabbitTemplate rabbitTemplate(ConnectionFactory connectionFactory) {  
 //注意 这个ConnectionFactory 是使用javaconfig方式配置连接的时候才需要传入的 如果是yml配置的连接的话是不需要的  
 RabbitTemplate template = new RabbitTemplate(connectionFactory)；  
 return template;  
}

@Component  
public class TestSend {  
  
 @Autowired  
 RabbitTemplate rabbitTemplate;  
  
 public void testSend() {  
 //至于为什么调用这个API 后面会解释  
 //参数介绍： 交换机名字，路由建， 消息内容  
 rabbitTemplate.convertAndSend("directExchange", "direct.key", "hello");  
 }  
}

我们只需要写一个类 然后交给spring 管理 在类里面注入RabbitTemplate 就可以直接调用api来发送消息了

接受消息：

这里我新建了一个项目（最好新建一个 当然 不新建也没关系） 来接收信息（之前的配置这里就不贴出来了 和上面基本一样），

@Component  
public class TestListener {  
  
  
 @RabbitListener(queues = "testQueue")  
 public void get(String message) throws Exception{  
 System.out.println(message);  
 }  
  
}

不出意外的话运行起来能看见效果， 这里就不贴效果图了

那么至此rabbitmq的一个快速开始 以及和spring boot整合 就完毕了， 下面开始会讲一些rabbitmq的一些高级特性 以及原理等