

仑+实战精

央速入门回顾

基础总结

高级特性总结

ŧπ

```
<> 代码块
                                                                                    索引目录
      connectionFactorv.setHost("127.0.0.1");
 1
      connectionFactory.setPort(5672);
 2
                                                                                    引入SpringBoot适配的
      connectionFactory.setPassword("guest");
 3
      connectionFactory.setUsername("guest");
                                                                                     Spring AMQP 特性
 4
 5
                                                                                    利用RebbitAdmin快速
      RabbitAdmin rabbitAdmin = new RabbitAdmin(connectionFactory);
 6
                                                                                     RebbitAdmin创建力
                                                                                     RebbitAdmin功能
      /*----*/
 8
                                                                                    利用RabbitAdmin简件
      Exchange exchange = new DirectExchange("exchange.order.restaurant");
 9
      rabbitAdmin.declareExchange(exchange);
                                                                                     RabbitAdmin声明式
10
                                                                                     RabbitAdmin声明式
11
      Queue queue = new Queue("queue.order");
12
                                                                                    利用RabbitTemplate
      rabbitAdmin.declareQueue(queue);
13
                                                                                     RabbitTemplate介结
14
                                                                                     RabbitTemplate简单
      Binding binding = new Binding(
15
                                                                                     RabbitTemplate设计
             "queue.order",
16
             Binding.DestinationType.QUEUE,
                                                                                    利用SimpleMessageL
17
             "exchange.order.restaurant",
                                                                                     SimpleMessageLis
18
             "key.order",
19
                                                                                     SimpleMessageLis
             null);
20
                                                                                    利用MessageListener
21
                                                                                     MessageListenerA
      rabbitAdmin.declareBinding(binding);
22
                                                                                     MessageListenerA
23
      /*----*/
                                                                                       简单模式: 实现ha
      exchange = new DirectExchange("exchange.order.deliveryman");
                                                                                       高阶模式: 自定义
25
      rabbitAdmin.declareExchange(exchange);
26
                                                                                    利用MessageConvert
      binding = new Binding(
27
                                                                                     MessageConverter
             "queue.order",
28
                                                                                     常用的MessageCor
             Binding.DestinationType.QUEUE,
29
                                                                                     自定义Me
             "exchange.order.deliveryman",
30
                                                                                              \Box
             "key.order",
                                                                                     如何使用.
31
             null):
                                                                                     如何使用
32
                                                                                              ?
     rabbitAdmin.declareBinding(binding);
33
                                                                                    利用Rabbit
34
                                                                                     RabbitLis
35
                                                                                     @Rabbit
      /*----*/
36
      exchange = new FanoutExchange("exchange.order.settlement");
37
      rabbitAdmin.declareExchange(exchange);
38
      exchange = new FanoutExchange("exchange.settlement.order");
39
      rabbitAdmin.declareExchange(exchange);
40
      binding = new Binding(
41
             "queue.order",
42
             Binding.DestinationType.QUEUE,
4.3
             "exchange.order.settlement",
44
             "key.order",
45
             null):
46
     rabbitAdmin.declareBinding(binding);
47
48
49
      /*----*/
50
      exchange = new TopicExchange("exchange.order.reward");
51
      rabbitAdmin.declareExchange(exchange);
      binding = new Binding(
53
             "queue.order",
54
             Binding.DestinationType.QUEUE,
55
             "exchange.order.reward",
56
             "key.order",
57
             null);
58
      rabbitAdmin.declareBinding(binding);
```

利用RabbitAdmin简化配置Rabbit服务流程





口 标记书签

奴程 Ξ

仑+实战精

央速入门回顾

基础总结

高级特性总结

耒群高可用

```
RabbitAdmin声明式配置的优点
```

- 将声明和创建工作分开,解耦多人工作
- 不需显式声明,减少代码量,减少Bug

RabbitAdmin声明式配置方法

将Exchange、Queue、Binding声明为Bean

public Binding binding2(){

return new Binding(

"queue.order",

"key.order",

null);

/*----*/

public Exchange exchange3(){

public Exchange exchange4(){

public Binding binding3(){

return new Binding(

▶ 意见反馈

"queue.order",

"key order"

Binding.DestinationType.QUEUE,

"exchange.order.deliveryman",

return new FanoutExchange("exchange.order.settlement");

return new FanoutExchange("exchange.settlement.order");

口 标记书签

Binding.DestinationType.QUEUE, "exchange.settlement.order",

♡ 收藏教程

26

2.7 2.8

29

30

31 32

33 34

3.5 36 37

38 39

40

41 42

43 44

45 46

47

48

49

}

索引目录

引入SpringBoot适配的

Spring AMQP 特性

利用RebbitAdmin快速

RebbitAdmin创建力

RebbitAdmin功能 利用RabbitAdmin简件

RabbitAdmin声明式

\dmin声明:

?

```
<> 代码块
                                                                                tTemplate
     /*----*/
                                                                                 RabbitTemplate介结
 2
     @Bean
                                                                                 RabbitTemplate简单
 3
     public Exchange exchange1(){
                                                                                 RabbitTemplate设计
 4
         return new DirectExchange("exchange.order.restaurant");
                                                                                利用SimpleMessageL
 5
     }
                                                                                 SimpleMessageList
     @Bean
 6
 7
     public Queue queue1(){
                                                                                 SimpleMessageLis
 8
         return new Queue("queue.order");
                                                                                利用MessageListener
 9
     }
                                                                                 MessageListenerA
     @Bean
1.0
                                                                                 MessageListenerA
11
     public Binding binding1(){
                                                                                   简单模式: 实现ha
         return new Binding(
12
                                                                                   高阶模式: 自定义
13
                "queue.order",
14
                Binding.DestinationType.QUEUE,
                                                                                利用MessageConvert
                "exchange.order.deliveryman",
15
                                                                                 MessageConverter
                "key.order",
16
                                                                                 常用的MessageCor
                null);
17
                                                                                 自定义Me
18
                                                                                 如何使用.
19
                                                                                 如何使用
     /*----*/
2.0
21
                                                                                利用Rabbit
22
     public Exchange exchange2(){
                                                                                 RabbitLis
23
         return new DirectExchange("exchange.order.deliveryman");
                                                                                 @Rabbit
2.4
     }
25
     @Bean
```

仑+实战精

央速入门回顾

基础总结

高级特性总结

ŧπ

 丰群高可用

```
52
 53
                                                                                     索引目录
       /*----*/
 54
 55
                                                                                     引入SpringBoot适配的
      public Exchange exchange5(){
 56
                                                                                       Spring AMQP 特性
 57
          return new TopicExchange("exchange.order.reward");
                                                                                     利用RebbitAdmin快速
 58
                                                                                       RebbitAdmin创建力
      @Bean
 59
      public Binding binding4(){
 60
                                                                                       RebbitAdmin功能
          return new Binding(
 61
                                                                                     利用RabbitAdmin简件
 62
                  "queue.order",
                                                                                       RabbitAdmin声明式
                  Binding.DestinationType.QUEUE,
 6.3
                                                                                       RabbitAdmin声明式
 64
                  "exchange.order.reward",
                                                                                     利用RabbitTemplate
                  "key.order",
                  null);
                                                                                       RabbitTemplate介结
 66
 67
                                                                                       RabbitTemplate简单
                                                                                       RabbitTemplate设计
再将RabbitAdmin声明为Bean
                                                                                     利用SimpleMessageL
```

```
C:---'-MessageLis
<> 代码块
                                                                                               √lessageList
                                                                                        利用MessageListener
 2
      public ConnectionFactory connectionFactory(){
                                                                                         MessageListenerA
          CachingConnectionFactory connectionFactory = new CachingConnectionFactory();
 3
                                                                                         MessageListenerA
 4
          connectionFactory.setHost("127.0.0.1");
                                                                                           简单模式: 实现ha
          connectionFactory.setPort(5672);
                                                                                           高阶模式: 自定义
          connectionFactory.setPassword("guest");
 6
                                                                                        利用MessageConvert
          connectionFactory.setUsername("guest");
 8
          connectionFactory.createConnection();
                                                                                         MessageConverter
 9
          return connectionFactory;
                                                                                         常用的MessageCor
      }
1.0
                                                                                         自定义Me
11
                                                                                         如何使用.
12
                                                                                         如何使用
13
      public RabbitAdmin rabbitAdmin(ConnectionFactory){
                                                                                                  ?
                                                                                        利用Rabbit
          RabbitAdmin rabbitAdmin = new RabbitAdmin(connectionFactory);
14
15
          rabbitAdmin.setAutoStartup(true);
                                                                                         RabbitLis
16
          return rabbitAdmin;
                                                                                         @Rabbit
```

Exchange、Queue、Binding即可自动创建

利用RabbitTemplate快速消息发送

RabbitTemplate**介绍**

17

RabbitTemplate与RestTemplate类似,使用了模板方法设计模式 RabbitTemplate提供了丰富的功能,方便消息收发 RabbitTemplate可以显式传入配置也可以隐式声明配置

RabbitTemplate简单使用方法

声明RabbitTemplate Bean

```
1  @Bean
2  RabbitTemplate rabbitTemplate(ConnectionFactory connectionFactory) {
3    RabbitTemplate rabbitTemplate = new RabbitTemplate(connectionFactory);
4    return rabbitTemplate;
5 }
```



□ 标记书签

奴程 Ξ

仑+实战精

央速入门回顾

基础总结

高级特性总结

耒群高可用

```
<> 代码块
       riessage message - new riessage(messagerosenu.gecbytes(), messageri opei ties),
                                                                                        索引目录
 1
       rabbitTemplate.send(
 2
               "exchange.order.restaurant",
 3
               "key.restaurant",
                                                                                        引入SpringBoot适配的
 4
               message,correlationData
                                                                                          Spring AMQP 特性
 5
       );
                                                                                        利用RebbitAdmin快速
                                                                                          RebbitAdmin创建力
使用convertAndSend方法发送String类型消息
                                                                                          RebbitAdmin功能
                                                                                                itAdmin简件
 <> 代码块
                                                                                          Nuvvivdmin声明記
       rabbitTemplate.send(
                                                                                          RabbitAdmin声明式
               "exchange.order.restaurant",
                                                                                        利用RabbitTemplate
 3
               "key.restaurant",
                                                                                          RabbitTemplate介绍
 4
               message
                                                                                          RabbitTemplate简单
       );
                                                                                          RabbitTemplate设计
                                                                                        利用SimpleMessageL
RabbitTemplate设置发送端确认、消息返回方法
                                                                                          SimpleMessageList
                                                                                          SimpleMessageList
RabbitTemplate设置setConfirmCallback, setReturnCallback
                                                                                        利用MessageListener
                                                                                                eListenerA
 <> 代码块
                                                                                          www.geListenerA
  1
       @Bean
                                                                                           简单模式: 实现ha
  2
       RabbitTemplate rabbitTemplate(ConnectionFactory connectionFactory) {
                                                                                           高阶模式: 自定义
           RabbitTemplate rabbitTemplate = new RabbitTemplate(connectionFactory);
  3
                                                                                        利用MessageConvert
           rabbitTemplate.setMandatory(true);
  4
                                                                                          MessageConverter
  5
                                                                                          常用的MessageCor
  6
           rabbitTemplate.setConfirmCallback((correlationData, ack, cause) ->
                   log.info("correlationData:{}, ack:{}, cause{}",
                                                                                          自定义Me
  8
                           correlationData,
                                                                                          如何使用.
  9
                           ack,
                                                                                          如何使用
                                                                                                   ?
 1.0
                           cause));
           和用Rabbit rabbitTemplate.setReturnCallback((message, replyCode, replyText, exchange, routingKe
 11
                                                                                          RabbitLis
 12
                   log.info(
                           "message:{}, replyCode:{}, replyText:{}, exchange:{}, routingKepRabbit
 1.3
 14
                           message,
                                                                                                   \odot
                           replyCode,
                           replyText,
 16
                           exchange,
 17
 18
                           routingKey));
 19
           return rabbitTemplate;
 20
       }
      4
发送消息时携带CorrelationData
 <> 代码块
 1
       CorrelationData correlationData = new CorrelationData();
 2
       correlationData.setId(orderPO.getId().toString());
 3
       rabbitTemplate.send(
               "exchange.order.restaurant",
 4
               "key.restaurant",
 5
 6
               message,correlationData
       );
```

利用SimpleMessageListenerContainer高效监听消息

SimpleMessageListenerContainer 特点



好程 :≡

仑+实战精

央速入门回顾

基础总结

高级特性总结

ŧπ

耒群高可用

```
设置同时监听多个队列、自动启动、自动配置RabbitMQ
设置消费者数量(最大数量、最小数量、批量消费)
设置消息确认模式、是否重回队列、异常捕获
设置是否独占、其他消费者属性等
设置具体的监听器、消息转换器等
支持动态设置,运行中修改监听器配置
```

SimpleMessageListenerContainer 使用方法

新建SimpleMessageListenerContainer Bean,并设置MessageListener

索引目录

引入SpringBoot适配 Spring AMQP 特性 利用RebbitAdmin快返

RebbitAdmin创建方

RebbitAdmin功能

利用RabbitAdmin简件

RabbitAdmin声明式

RabbitAdmin声明式

```
<> 代码块
                                                                                                                                                                                                                                                               tTemplate
                                                                                                                                                                                                                                               RabbitTemplate介绍
                @Bean
   1
                public SimpleMessageListenerContainer messageListenerContainer(ConnectionFactory connect
   2
                          {\tt Simple Message Listener Container \ message Listener Container \ = \ new \ Simple Message Listener Container} = new \ {\tt Simple Message Listener Container}
   3
                          messageListenerContainer.setQueueNames("queue.order");
                                                                                                                                                                                                                                           利用SimpleMessageL
                          messageListenerContainer.setConcurrentConsumers(1);
   5
                                                                                                                                                                                                                                               SimpleMessageLis
                           messageListenerContainer.setMaxConcurrentConsumers(3);
   6
                                                                                                                                                                                                                                               SimpleMessageLis
                           messageListenerContainer.setAcknowledgeMode(AcknowledgeMode.AUTO);
                                                                                                                                                                                                                                            利用MessageListener
   8
                           messageListenerContainer.setMessageListener(new MessageListener() {
                                                                                                                                                                                                                                               MessageListenerA
   9
                                     @Override
                                                                                                                                                                                                                                               MessageListenerA
1 0
                                     public void onMessage(Message message) {
                                                log.info("message:{}", message);
                                                                                                                                                                                                                                                    简单模式: 实现ha
12
                                                                                                                                                                                                                                                    高阶模式: 自定义
1.3
                          });
                                                                                                                                                                                                                                           利用MessageConvert
                           messageListenerContainer.setPrefetchCount(2);
                                                                                                                                                                                                                                                MessageConverter
                           {\tt messageListenerContainer.setAcknowledgeMode(AcknowledgeMode.MANUAL);}
                                                                                                                                                                                                                                               常用的MessageCo
                           messageListenerContainer.setMessageListener(new ChannelAwareMessageListener()
16
                                                                                                                                                                                                                                                自定义Me
17
                                     public void onMessage(Message message, Channel channel) throws Exception {如何使用.
18
                                                 channel.basicAck(message.getMessageProperties().getDeliveryTag(),false如何使用
19
2.0
                                                                                                                                                                                                                                           利用Rabbit
21
                                                                                                                                                                                                                                               RabbitLis
                           return messageListenerContainer;
                                                                                                                                                                                                                                                @Rabbit
23
```

利用MessageListenerAdapter自定义消息监听

MessageListenerAdapter消息监听适配器特点

采用了适配器设计模式

解决业务逻辑代码无法修改的问题

MessageListenerAdapter使用方法

简单模式: 实现handleMessage方法

将目标业务service注入MessageListenerAdapter

奴程 ≔

仑+实战精

央速入门回顾

基础总结

高级特性总结

旧

耒群高可用

```
messageListenerContainer.setMaxConcurrentConsumers(5);
```

```
10
          messageListenerContainer.setAcknowledgeMode(AcknowledgeMode.AUTO);
                                                                                        索引日录
11
12
          MessageListenerAdapter listenerAdapter = new MessageListenerAdapter();
                                                                                        引入SpringBoot适配的
          listenerAdapter.setDelegate(orderMessageService);
1.3
                                                                                          Spring AMQP 特性
14
                                                                                        利用RebbitAdmin快速
1.5
          Map<String, String> methodMap = new HashMap<>(8);
                                                                                          RebbitAdmin创建力
16
          messageListenerContainer.setMessageListener(listenerAdapter);
17
          return messageListenerContainer;
                                                                                          RebbitAdmin功能
18
                                                                                        利用RabbitAdmin简件
```

目标service实现handleMessage方法

9

```
itTemplate
<> 代码块
                                                                                               ....emplate介绍
      @Service
1
                                                                                               RabbitTemplate简单
2
      public class OrderMessageService {
                                                                                               RabbitTemplate设计
3
                                                                                             利用SimpleMessageL
                                                                                               SimpleMessageLis
5
          public void handleMessage1(byte[] messageBody) throws IOException {
                                                                                               SimpleMessageLis
               //业务逻辑
6
                                                                                             利用MessageListener
8
          }
                                                                                               MessageListenerA
```

高阶模式: 自定义"队列名→方法名"映射关系

将目标业务service注入MessageListenerAdapter,并设置自定义"队列名→方法名"映射关系

```
eConverter
<> 代码块
                                                                                            רבונו, בחירו, בחירו, בחירו, בחירו
 1
                                                                                            自定义Me
      public SimpleMessageListenerContainer messageListenerContainer(
                                                                                            如何使用.
 3
              @Autowired ConnectionFactory connectionFactory
                                                                                            如何使用
 4
      ) {
                                                                                          利用Rabbit
 5
          SimpleMessageListenerContainer messageListenerContainer =
                                                                                            RabbitLis
 6
                  new SimpleMessageListenerContainer(connectionFactory);
 7
          messageListenerContainer.setQueueNames("queue.order");
                                                                                            @Rabbit
 8
          messageListenerContainer.setConcurrentConsumers(3);
          messageListenerContainer.setMaxConcurrentConsumers(5):
 9
10
          messageListenerContainer.setAcknowledgeMode(AcknowledgeMode.AUTO);
11
          MessageListenerAdapter listenerAdapter = new MessageListenerAdapter();
12
          listenerAdapter.setDelegate(orderMessageService);
1.3
14
          Map<String, String> methodMap = new HashMap<>(8);
15
          methodMap.put("queue.order", "handleMessage1");
16
          listenerAdapter.setQueueOrTagToMethodName(methodMap);
17
          messageListenerContainer.setMessageListener(listenerAdapter);
18
19
          return messageListenerContainer;
2.0
```

目标service实现自定义方法

```
<> 代码块
1
      @Service
      public class OrderMessageService {
3
4
5
         public void handleMessage(byte[] messageBody) throws IOException {
6
              //业务逻辑
8
```



RabbitAdmin声明式

RabbitAdmin声明云

MessageListenerA

利用MessageConvert

简单模式:实现ha 高阶模式: 自定义

 \Box

仑+实战精

央速入门回顾

基础总结

高级特性总结

in

利用MessageConverter高效处理消息

引入SpringBoot适配的

索引日录

之前收发消息时,使用了Byte[]数组作为消息体,但是编写业务逻辑时,需要使用Java对象的 AMQP 特性 MessageConverter用来在收发消息时自动转换消息 利用RebbitAdmin快速

RebbitAdmin创建方 RebbitAdmin功能

常用的MessageConverter

MessageConverter的意义

Jackson2JsonMessageConverter是最常用的MessageConverter

如何使用Jackson2JsonMessageConverter

给MessageListenerAdapter设置MessageConverter

用来转换Json格式消息

配合ClassMapper可以直接转换为POJO对象

利用RabbitAdmin简件

RabbitAdmin声明云

RabbitAdmin声明云

利用RabbitTemplate

RabbitTemplate介绍

自定义MessageConverter

实现MessageConverter接口

<> 代码块

@Bean

) {

1

2

3

5

6

8

9

10

11

12 13

14

重写toMessage、fromMessage方法

RabbitTemplate简单

RabbitTemplate设 利用SimpleMessageL

SimpleMessageLis

SimpleMessageLis

利用MessageListener

MessageListenerA

MessageListenerA

简单模式: 实现ha

```
試:自定义
利用MessageConverle
MessageConverter
常用的MessageCor
自定义M·
」
如何使用.
```

如何使用。 如何使用(利用Rabbit

RabbitLis @Rabbit

0

MessageListenerAdapter messageListenerAdapter = new MessageListenerAdapter(orderMess Jackson2JsonMessageConverter messageConverter = new Jackson2JsonMessageConverter(); messageListenerAdapter.setMessageConverter(messageConverter);

messageListenerAdapter.setMessageConverter(messageConverter);
messageListenerContainer.setMessageListener(messageListenerAdapter);

public SimpleMessageListenerContainer messageListenerContainer(

SimpleMessageListenerContainer messageListenerContainer =

new SimpleMessageListenerContainer(connectionFactory);

messageListenerContainer.setAcknowledgeMode(AcknowledgeMode.AUTO);

@Autowired ConnectionFactory connectionFactory

messageListenerContainer.setQueueNames("queue.order");

messageListenerContainer.setMaxConcurrentConsumers(5);

messageListenerContainer.setConcurrentConsumers(3);

messageListenerContainer.setPrefetchCount(1);

17 return messageListenerContainer;
18 }

在业务中使用Map接收消息body

如何使用ClassMapper

给Jackson2JsonMessageConverter设置ClassMapper



♡ 收藏教程

□ 标记书签

仑+实战精

央速入门回顾

基础总结

高级特性总结

П

```
<> 代码块
                                                                                         深51日求
  1
       @Bean
       public SimpleMessageListenerContainer messageListenerContainer(
                                                                                         引入SpringBoot适配的
  3
               @Autowired ConnectionFactory connectionFactory
                                                                                           Spring AMQP 特性
  4
       ) {
                                                                                         利用RebbitAdmin快速
  5
           SimpleMessageListenerContainer messageListenerContainer =
                                                                                           RebbitAdmin创建力
                   new SimpleMessageListenerContainer(connectionFactory);
  6
           messageListenerContainer.setOueueNames("queue.order");
                                                                                           RebbitAdmin功能
  8
           messageListenerContainer.setConcurrentConsumers(3);
                                                                                         利用RabbitAdmin简件
  9
           messageListenerContainer.setMaxConcurrentConsumers(5);
                                                                                           RabbitAdmin声明式
           messageListenerContainer.setAcknowledgeMode(AcknowledgeMode.AUTO);
 10
                                                                                           RabbitAdmin声明式
 11
                                                                                         利用RabbitTemplate
 12
           messageListenerContainer.setPrefetchCount(1);
 13
                                                                                           RabbitTemplate介绍
           MessageListenerAdapter messageListenerAdapter = new MessageListenerAdapter(ord解ងអាក្emplate简直
 14
           Jackson2JsonMessageConverter messageConverter = new Jackson2JsonMessageConverter()
 15
           messageConverter.setClassMapper(new ClassMapper() {
                                                                                         利用SimpleMessageL
 17
               @Override
                                                                                           SimpleMessageLis
               public void fromClass(Class<?> clazz, MessageProperties properties) {
 1.8
                                                                                           SimpleMessageLis
 19
                                                                                         利用MessageListener
 21
                                                                                           MessageListenerA
 22
               @Override
                                                                                           MessageListenerA
               public Class<?> toClass(MessageProperties properties) {
 23
                                                                                            简单模式: 实现ha
                   return OrderMessageDTO.class;
                                                                                            高阶模式: 自定义
 25
                                                                                         利用MessageConvert
 26
 2.7
           messageListenerAdapter.setMessageConverter(messageConverter);
                                                                                           MessageConverter
           messageListenerContainer.setMessageListener(messageListenerAdapter);
 2.8
                                                                                           常用的MessageCor
 29
           return messageListenerContainer;
                                                                                           自定义Me
 30
                                                                                           如何使用.
                                                                                           如何使用
                                                                                         利用Rabbit
在业务中使用lava目标对象直接接收消息body
                                                                                           RabbitLis
 <> 代码块
       @Service
 1
       public class OrderMessageService {
           public void handleMessage(OrderMessageDTO orderMessageDTO) {
 3
 Δ
                //业务逻辑
 5
        }
 6
```

利用RabbitListener快速实现消息处理器

RabbitListener特点

RabbitListener是SpringBoot架构中监听消息的"终极方案" RabbitListener使用注解声明,对业务代码无侵入 RabbitListener可以在SpringBoot配置文件中进行配置

@RabbitListener注解的使用

@RabbitListener是一个组合注解,可以嵌套以下注解:

@Exchange: 自动声明Exchange

@Queue: 自动声明队列

@QueueBinding: 自动声明绑定关系

```
✔ 意见反馈 🗸 收藏教程 🗘 标记书签
```

好程 :≡

仑+实战精

央速入门回顾

基础总结

高级特性总结

1

≢群入门

```
<> 代码块
                                                                                        察51日求
  1
          @RabbitListener(
  2
                     containerFactory = "rabbitListenerContainerFactory",
       //
                                                                                        引入SpringBoot适配的
  3
       //
                     queues = "queue.order",
                                                                                          Spring AMQP 特性
                     admin = "rabbitAdmin",
       //
  4
                                                                                        利用RebbitAdmin快速
  5
                   bindings = {
                                                                                          RebbitAdmin创建力
  6
                           @QueueBinding(
                                   value = @Queue(name = "${imooc.order-queue}",
                                                                                          RebbitAdmin功能
  8
                                           arguments = {
                                                                                        利用RabbitAdmin简件
  9
                                                   // @Argument(name =
                                                                                          RabbitAdmin声明式
                                                   // "x-message-ttl", value =
 10
                                                                                          RabbitAdmin声明式
                                                       "1000", type = "java.lang
 11
                                                                                        利用RabbitTemplate
 12
                                                       .Integer"),
 13
                                                       @Argument(name =
                                                                                          RabbitTemplate介绍
                                                       "x-dead-letter-exchange",
 14
                                                                                          RabbitTemplate简单
                                                   // value = "aaaaa"),
 15
                                                                                          RabbitTemplate设计
 16
                                                       @Argument(name =
                                                                                        利用SimpleMessageL
 17
                                                   // "x-dead-letter-routing-key", value
                                                                                          SimpleMessageLis
                                           }),
 1.8
                                   exchange = @Exchange(name = "exchange.order.restaurant'simpleMessageList
 19
                                   key = "key.order"
                                                                                        利用MessageListener
 20
 21
                           ),
                                                                                          MessageListenerA
 22
                           @OueueBinding(
                                                                                          MessageListenerA
 23
                                   value = @Queue(name = "queue.order"),
                                                                                           简单模式: 实现ha
                                   exchange = @Exchange(name = "exchange.order.deliveryman"
                                                                                           高阶模式: 自定义
                                   key = "key.order"
 25
                                                                                        利用MessageConvert
 26
 27
                           @QueueBinding(
                                                                                          MessageConverter
                                   value = @Queue(name = "queue.order"),
 2.8
                                                                                          常用的MessageCor
                                   exchange = @Exchange(name = "exchange.settlement.order"
 29
                                                                                                   \Box
                                   key = "key.order"
 30
                                                                                          如何使用.
 31
                           ),
                                                                                          如何使用
                           @QueueBinding(
 32
                                                                                                   ?
                                                                                        利用Rabbit
 33
                                   value = @Queue(name = "queue.order"),
                                   exchange = @Exchange(name = "exchange.order.reward", tyRabbitLis
 34
                                   key = "key.order"
 35
                                                                                          @Rabbit
 36
                           )
                                                                                                   \odot
 37
                   }
 38
           public void handleMessage(@Payload Message message) throws IOException {
 39
               //业务逻辑
 40
 41
 42
配合配置文件,可以自动生成RabbitAdmin等组件
 <> 代码块
       spring.rabbitmq.username=guest
 1
 2
       spring.rabbitmq.password=guest
 3
       spring.rabbitmq.addresses=127.0.0.1
       spring.rabbitmq.port=6379
               第四章RabbitMO高级特性总结 ◆ 上一节
                                                   下一节 ▶ 第六音RabbitMO集群入门
                                        ▶ 我要提出意见反馈
```