

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/spring/MyConsumer.java - MyEclipse Enterprise Workbench

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help

Package Explorer Type Hierarchy

15.spring集成rabbitmq client

SpringMain.java context.xml MyConsumer.java

```
public class MyConsumer {  
    //具体执行业务的方法  
    public void listen(String foo) {  
        System.out.println("消费者: " + foo);  
    }  
}
```

Tasks Web Browser Servers Progress JUnit Console

<terminated> SpringMain [Java Application] C:\development\dev\develop\jdk1.7\bin\javaw.exe (2018-1-29 下午1:59:43)

再次双击继续播放

00:03:09

Writable Smart Insert 1:1

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/spring/SpringMain.java - MyEclipse Enterprise Workbench

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help

Package Explorer Type Hierarchy

15.spring集成rabbitmq client

SpringMain.java context.xml MyConsumer.java

```
import org.springframework.amqp.rabbit.core.RabbitTemplate;  
  
public class SpringMain {  
    public static void main(final String... args) throws Exception {  
        AbstractApplicationContext ctx = new ClassPathXmlApplicationContext("classpath:context.xml");  
        //RabbitMQ 模板  
        RabbitTemplate template = ctx.getBean(RabbitTemplate.class);  
        //发送消息  
        template.convertAndSend("Hello, world!");  
        Thread.sleep(1000); // 休眠1秒  
        ctx.destroy(); //容器销毁  
    }  
}
```

Tasks Web Browser Servers Progress JUnit Console

<terminated> SpringMain [Java Application] C:\development\dev\develop\jdk1.7\bin\javaw.exe (2018-1-29 下午1:59:43)

00:02:12

Writable Smart Insert 1:1

MyEclipse Java Enterprise - myrabbitmq/src/main/resources/context.xml - MyEclipse Enterprise Workbench

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/rabbit
http://www.springframework.org/schema/rabbit/spring-rabbit-1.7.xsd
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd">

    <!-- 1.定义RabbitMQ的连接工厂 -->
    <rabbit:connection-factory id="connectionFactory"
        host="127.0.0.1" port="5672" username="user_mmr" password="123"
        virtual-host="/vhost_mmr" />

    <!-- 2.定义Rabbit模板，指定连接工厂以及定义exchange -->
    <rabbit:template id="amqpTemplate" connection-factory="connectionFactory" exchange="fanoutExchange" />
    <!-- MQ的管理，包括队列、交换器声明等 -->
    <rabbit:admin connection-factory="connectionFactory" />

    <!-- 定义队列，自动声明 -->
    <rabbit:queue name="myQueue" auto-declare="true" durable="true"/>

    <!-- 定义交换器，自动声明 -->
    <rabbit:fanout-exchange name="fanoutExchange" auto-declare="true">
        <rabbit:bindings>
            <rabbit:binding queue="myQueue"/>
        </rabbit:bindings>
    </rabbit:fanout-exchange>

    <!-- 队列监听 -->
    <rabbit:listener-container connection-factory="connectionFactory">
        <rabbit:listener ref="foo" method="listen" queue-names="myQueue" />
    </rabbit:listener-container>

    <!-- 消费者 -->
    <bean id="foo" class="com.mmr.rabbitmq.spring.MyConsumer" />
</beans>

```

00:03:20 00

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/confirm/Recv.java - MyEclipse Enterprise Workbench

14.rabbitmq消息确认机制之confirm异步

```

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help

```

Package Explorer

- Other Projects
- servlet
  - demo1
  - demo2
- myrabbitmq
  - src/main/java
    - com.mmr.rabbitmq
      - confirm
        - Recv.java
        - Send1.java
        - Send2.java
        - Send3.java
      - ps
      - routing
      - simple
      - topic
    - tx
      - TxRecv.java
      - TxSend.java
    - util
    - work
    - workfair
  - src/main/resources
  - src/test/java
  - src/test/resources
- JRE System Library [J2SE-1.5]
- Maven Dependencies
- src
- target
- pom.xml

```

import java.io.IOException;
public class Recv {
    private static final String QUEUE_NAME="test_queue_confirm";
    public static void main(String[] args) throws IOException, TimeoutException {
        Connection connection = ConnectionUtils.getConnection();
        Channel channel = connection.createChannel();
        channel.queueDeclare(QUEUE_NAME,false,false,false,null);
        channel.basicConsume(QUEUE_NAME, true,new DefaultConsumer(channel){
            @Override
            public void handleDelivery(String consumerTag, Envelope envelope,
                                       BasicProperties properties, byte[] body) throws IOException {
                System.out.println("recv[confirm] msg:"+new String(body,"utf-8"));
            }
        });
    }
}

```

Tasks Web Browser Console Servers Progress JUnit

No consoles to display at this time.

00:07:27 00

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/confirm/Send3.java - MyEclipse Enterprise Workbench

```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
```

## 14.rabbitmq消息确认机制之confirm异步

```
1 package com.mmr.rabbitmq.confirm;
2
3 import java.io.IOException;
4
5 public class Send3 {
6     private static final String QUEUE_NAME="test_queue_confirm";
7
8     public static void main(String[] args) throws IOException, TimeoutException {
9         Connection connection = ConnectionUtils.getConnection();
10        Channel channel = connection.createChannel();
11        channel.queueDeclare(QUEUE_NAME,false,false,null);
12
13        //生产者调用confirmSelect 将channel设置为confirm模式 注意
14        channel.confirmSelect();
15
16        final SortedSet<Long> confirmSet= Collections.synchronizedSortedSet(new TreeSet<Long>());
17
18        channel.addConfirmListener(new ConfirmListener() {
19            //handleNack
20            public void handleNack(long deliveryTag, boolean multiple)
21                throws IOException {
22                if(multiple){
23                    System.out.println("----handleNack-----multiple");
24                    confirmSet.headSet(deliveryTag+1).clear();
25                }else{
26                    System.out.println("----handleNack-----multiple false");
27                    confirmSet.remove(deliveryTag);
28                }
29            }
30
31            //没有问题的handleAck
32            public void handleAck(long deliveryTag, boolean multiple)
33                throws IOException {
34                if(multiple){
35                    System.out.println("----handleAck----multiple");
36                    confirmSet.headSet(deliveryTag+1).clear();
37                }else{
38                    System.out.println("----handleAck----multiple false");
39                    confirmSet.remove(deliveryTag);
40                }
41            }
42        });
43
44        String msgStr="ssssss";
45
46        channel.basicPublish("",QUEUE_NAME,null,getBytes(msgStr));
47    }
48}
```

00:03:19 0

Writable Smart Insert 20 : 37

rabbitmqMQ笔记.docx - Microsoft Word

```
文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图
```

## 14.rabbitmq消息确认机制之confirm异步

### • 异步模式

Channel 对象提供的 ConfirmListener() 回调方法只包含 deliveryTag (当前 Channel 发出的消息序号)，我们需要自己为每一个 Channel 维护一个 unconfirm 的消息序号集合，每 publish 一条数据，集合中元素加 1，每回调一次 handleAck 方法，unconfirm 集合删掉相应的一条 (multiple=false) 或多条 (multiple=true) 记录。从程序运行效率上看，这个 unconfirm 集合最好采用有序集合 SortedSet 存储结构。

第 34 页, 共 35 页 3881 个字 中文(中国)

### 13.rabbitmq消息确认机制之confirm串行

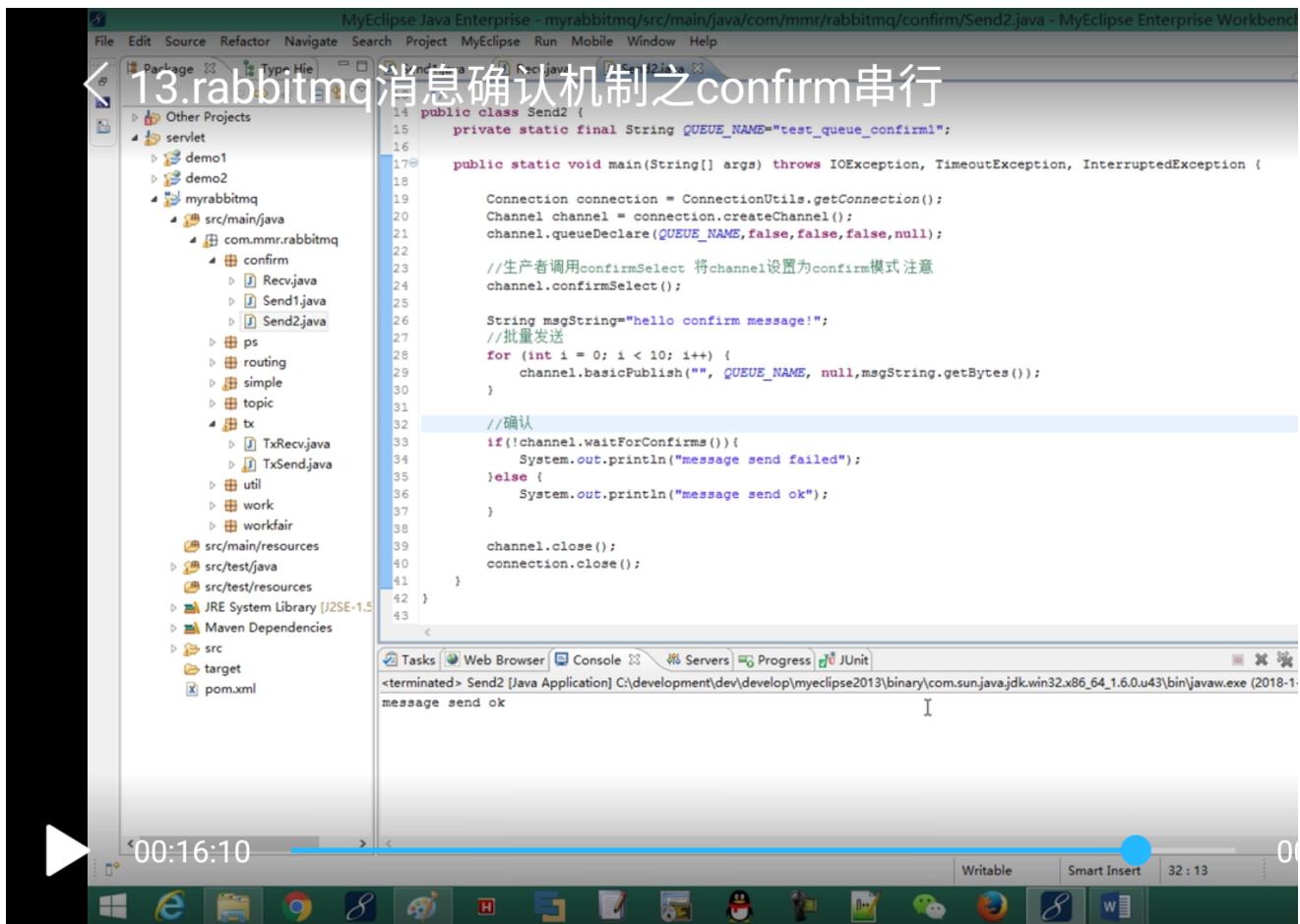
```
MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/confirm/Send2.java - MyEclipse Enterprise Workbench
```

```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
```

```
14 public class Send2 {
15     private static final String QUEUE_NAME="test_queue_confirm1";
16
17     public static void main(String[] args) throws IOException, TimeoutException, InterruptedException {
18
19         Connection connection = ConnectionUtils.getConnection();
20         Channel channel = connection.createChannel();
21         channel.queueDeclare(QUEUE_NAME,false,false,false,null);
22
23         //生产者调用confirmSelect 将channel设置为confirm模式注意
24         channel.confirmSelect();
25
26         String msgString="hello confirm message!";
27         //批量发送
28         for (int i = 0; i < 10; i++) {
29             channel.basicPublish("", QUEUE_NAME, null, msgString.getBytes());
30         }
31
32         //确认
33         if(!channel.waitForConfirms()){
34             System.out.println("message send failed");
35         }else {
36             System.out.println("message send ok");
37         }
38
39         channel.close();
40         connection.close();
41     }
42 }
43
```

```
Tasks Web Browser Console Servers Progress JUnit
<terminated> Send2 [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86_64_1.6.0.u43\bin\javaw.exe (2018-1-29 下午 00:16:10)
message send ok
```

```
00:16:10 Writable Smart Insert 32:13
```



### 13.rabbitmq消息确认机制之confirm串行

```
MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/confirm/Send1.java - MyEclipse Enterprise Workbench
```

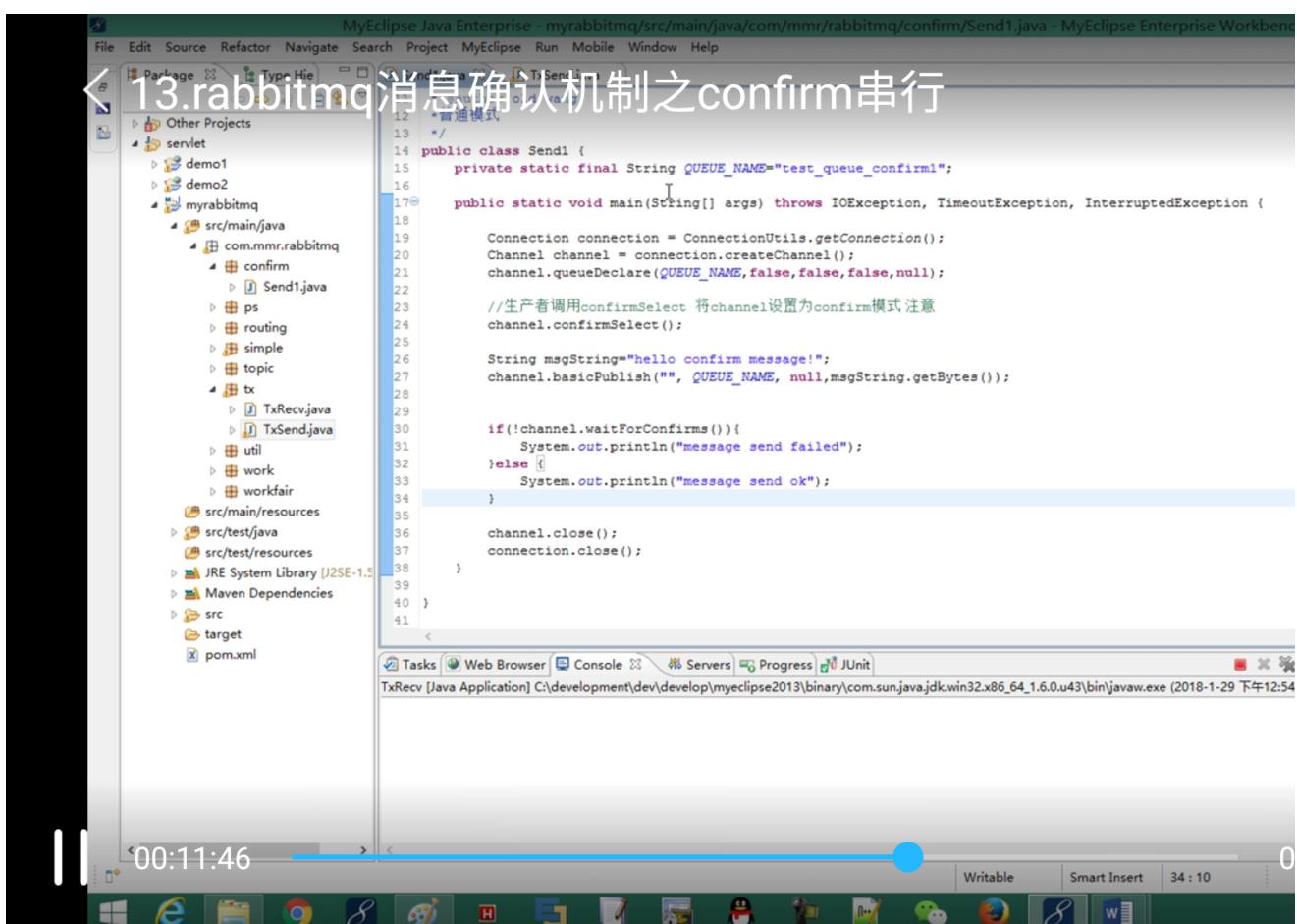
```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
```

```
12 /*普通模式
13 */
14 public class Send1 {
15     private static final String QUEUE_NAME="test_queue_confirm1";
16
17     public static void main(String[] args) throws IOException, TimeoutException, InterruptedException {
18
19         Connection connection = ConnectionUtils.getConnection();
20         Channel channel = connection.createChannel();
21         channel.queueDeclare(QUEUE_NAME,false,false,false,null);
22
23         //生产者调用confirmSelect 将channel设置为confirm模式注意
24         channel.confirmSelect();
25
26         String msgString="hello confirm message!";
27         channel.basicPublish("", QUEUE_NAME, null, msgString.getBytes());
28
29
30         if(!channel.waitForConfirms()){
31             System.out.println("message send failed");
32         }else {
33             System.out.println("message send ok");
34         }
35
36         channel.close();
37         connection.close();
38     }
39
40 }
```

```
Tasks Web Browser Console Servers Progress JUnit
TxRecv [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86_64_1.6.0.u43\bin\javaw.exe (2018-1-29 下午 00:11:46)

```

```
00:11:46 Writable Smart Insert 34:10
```



13.rabbitmq消息确认机制之confirm串行

```

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/confirm/Send1.java - MyEclipse Enterprise Workbench
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
13.rabbitmq消息确认机制之confirm串行
< Package Explorer > Type Hierarchy < Screenshot > Send1.java
1 Other Projects
2 < servlet
3   < demo1
4   < demo2
5   < myrabbitmq
6     < src/main/java
7       < com.mmr.rabbitmq
8         < confirm
9           < Send1.java
10        < ps
11        < routing
12        < simple
13        < topic
14        < tx
15          < TxRecv.java
16          < TxSend.java
17        < util
18        < work
19        < workfair
20      < src/main/resources
21      < src/test/java
22      < src/test/resources
23      < JRE System Library [J2SE-1.5]
24      < Maven Dependencies
25      < src
26      < target
27      < pom.xml
28
29
10 /**
11  * @author old wang
12  * 普通模式
13  */
14 public class Send1 {
15     private static final String QUEUE_NAME="test_queue_confirm1";
16
17     public static void main(String[] args) throws IOException, TimeoutException {
18
19         Connection connection = ConnectionUtils.getConnection();
20         Channel channel = connection.createChannel();
21         channel.queueDeclare(QUEUE_NAME,false,false,null);
22
23         //生产者调用confirmSelect 将channel设置为confirm模式 注意
24         channel.confirmSelect();
25
26     }
27
28 }
29

```

Tasks Web Browser Console Servers Progress JUnit

TxRecv [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86\_64\_1.6.0.u43\bin\javaw.exe (2018-1-29 下午12:54)

00:09:35

Writable Smart Insert 24 : 33

rabbitmqMQ笔记.docx - Microsoft Word

**· 生产者端 confirm 模式的实现原理**

生产者将信道设置成 confirm 模式，一旦信道进入 confirm 模式，所有在该信道上面发布的消息都会被指派一个唯一的 ID(从 1 开始)，一旦消息被投递到所有匹配的队列之后，broker 就会发送一个确认给生产者(包含消息的唯一 ID)，这就使得生产者知道消息已经正确到达目的队列了，如果消息和队列是可持久化的，那么确认消息会将消息写入磁盘之后发出，broker 回传给生产者的确认消息中 deliver-tag 域包含了确认消息的序列号，此外 broker 也可以设置 basic.ack 的 multiple 域，表示到这个序列号之前的所有消息都已经得到了处理。

· 生产者端 confirm 模式的实现...

消息应答  
消息的持久化  
4. 订阅模式publish/subscribe  
模型  
生产者  
消费者1  
消费者2  
6. Exchange(交换机, 转发器)  
Fanout(不处理路由键)  
Direct (处理路由键)  
Topic exchange  
路由模式  
模型  
生产者  
消费者1  
消费者2  
Topic  
模型  
消费者1  
消费者2  
Rabbitmq的消息确认机制(事务...  
事务机制  
生产者  
消费者  
Confirm模式  
生产者端confirm模式的实现...

第 32 页, 共 32 页 3536 个字 英语(美国)

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/tx/TxSend.java - MyEclipse Enterprise Workbench

## 12.rabbitmq消息确认机制之事务机制

```

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help

1 package com.mmr.rabbitmq;
2
3 import com.rabbitmq.client.Channel;
4 import com.rabbitmq.client.Connection;
5
6 public class TxSend {
7     private static final String QUEUE_NAME="test_queue_tx";
8
9     public static void main(String[] args) throws IOException, TimeoutException {
10
11         Connection connection = ConnectionUtils.getConnection();
12         Channel channel = connection.createChannel();
13         channel.queueDeclare(QUEUE_NAME,false,false,null);
14         String msgString="hello tx message!";
15
16         try {
17             channel.txSelect();
18             channel.basicPublish("", QUEUE_NAME, null, msgString.getBytes());
19             channel.txCommit();
20         } catch (Exception e) {
21             channel.txRollback();
22             System.out.println(" send message txRollback");
23         }
24
25         channel.close();
26         connection.close();
27     }
28
29
30
31
32
33
34
35
36

```

Tasks Web Browser Console Servers Progress JUnit

No consoles to display at this time.

00:11:08

Writable Smart Insert 16 : 9

rabbitmqMQ笔记.docx - Microsoft Word

## 12.rabbitmq消息确认机制之事务机制

• Rabbitmq 的消息确认机制(事务+confirm).

在 rabbitmq 中 我们可以通过持久化数据 解决 rabbitmq 服务器异常 的数据丢失问题,

问题:生产者将消息发送出去之后,消息到底有没有到达 rabbitmq 服务器,默认的情况是不知道的;

两种方式:

- AMQP 实现了事务机制,
- Confirm 模式

事务|

第 30 页, 共 30 页 3133 个字 中文(中国)

11.topic主题模式

```
MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/topic/Recv2.java - MyEclipse Enterprise Workbench
```

```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
```

```
Package Type Hierarchy Java Recv1.java * Recv2.java * java.io.IOException;
```

```
13 public class Recv2 {  
14     private static final String EXCHANGE_NAME = "test_exchange_topic";  
15     private static final String QUEUE_NAME = "test_queue_topic_2";  
16  
17     public static void main(String[] args) throws IOException, TimeoutException {  
18         Connection connection = ConnectionUtils.getConnection();  
19         final Channel channel = connection.createChannel();  
20  
21         channel.queueDeclare(QUEUE_NAME, false, false, false, null);  
22         channel.queueBind(QUEUE_NAME, EXCHANGE_NAME, "goods.#");  
23  
24         channel.basicQos(1);  
25  
26         //定义一个消费者  
27         Consumer consumer=new DefaultConsumer(channel){  
28             //消息到达触发这个方法  
29             @Override  
30             public void handleDelivery(String consumerTag, Envelope envelope,  
31                                         BasicProperties properties, byte[] body) throws IOException {  
32                 String msg=new String(body,"utf-8");  
33                 System.out.println("[2] Recv msg:"+msg);  
34  
35                 try {  
36                     Thread.sleep(2000);  
37                 } catch (InterruptedException e) {  
38                     e.printStackTrace();  
39                 }finally{  
40                     System.out.println("[2] done ");  
41                     channel.basicAck(envelope.getDeliveryTag(), false);  
42                 }  
43             }  
44         };  
45         channel.basicConsume(QUEUE_NAME, true, consumer);  
46     }  
47 };
```

```
Tasks Web Browser Console Servers Progress
```

```
Recv1 (3) [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86_64_1.6.0.u43\bin\javaw.exe (2018-1-28 下午8:1)
```

```
00:10:23
```

```
Writable Smart Insert 16 : 65
```

11.topic主题模式

```
MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/topic/Recv2.java - MyEclipse Enterprise Workbench
```

```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
```

```
Package Type Hierarchy Java Recv1.java * Recv2.java * java.io.IOException;
```

```
13 public class Recv2 {  
14     private static final String EXCHANGE_NAME = "test_exchange_topic";  
15     private static final String QUEUE_NAME = "test_queue_topic_2";  
16  
17     public static void main(String[] args) throws IOException, TimeoutException {  
18         Connection connection = ConnectionUtils.getConnection();  
19         final Channel channel = connection.createChannel();  
20  
21         channel.queueDeclare(QUEUE_NAME, false, false, false, null);  
22         channel.queueBind(QUEUE_NAME, EXCHANGE_NAME, "goods.#");  
23  
24         channel.basicQos(1);  
25  
26         //定义一个消费者  
27         Consumer consumer=new DefaultConsumer(channel){  
28             //消息到达触发这个方法  
29             @Override  
30             public void handleDelivery(String consumerTag, Envelope envelope,  
31                                         BasicProperties properties, byte[] body) throws IOException {  
32                 String msg=new String(body,"utf-8");  
33                 System.out.println("[2] Recv msg:"+msg);  
34  
35                 try {  
36                     Thread.sleep(2000);  
37                 } catch (InterruptedException e) {  
38                     e.printStackTrace();  
39                 }finally{  
40                     System.out.println("[2] done ");  
41                     channel.basicAck(envelope.getDeliveryTag(), false);  
42                 }  
43             }  
44         };  
45         channel.basicConsume(QUEUE_NAME, true, consumer);  
46     }  
47 };
```

```
Tasks Web Browser Console Servers Progress
```

```
Recv1 (3) [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86_64_1.6.0.u43\bin\javaw.exe (2018-1-28 下午8:1)
```

```
00:10:23
```

```
Writable Smart Insert 16 : 65
```

再次双击继续播放

11.topic主题模式

```

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/topic/Recv1.java - MyEclipse Enterprise Workbench
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
Package Type Hierarchy Java Recv1.java
1 package com.mmr.rabbitmq.topic;
2
3 import java.io.IOException;
4
5 public class Recv1 {
6     private static final String EXCHANGE_NAME = "test_exchange_topic";
7     private static final String QUEUE_NAME = "test_queue_topic_1";
8
9     public static void main(String[] args) throws IOException, TimeoutException {
10
11         Connection connection = ConnectionUtils.getConnection();
12         final Channel channel = connection.createChannel();
13
14         channel.queueDeclare(QUEUE_NAME, false, false, false, null);
15
16         channel.queueBind(QUEUE_NAME, EXCHANGE_NAME, "goods.add");
17
18         channel.basicQos(1);
19
20         //定义一个消费者
21         Consumer consumer=new DefaultConsumer(channel){
22             //消息到达触发这个方法
23             @Override
24             public void handleDelivery(String consumerTag, Envelope envelope,
25                                         BasicProperties properties, byte[] body) throws IOException {
26
27                 String msg=new String(body,"utf-8");
28                 System.out.println("[1] Recv msg:"+msg);
29
30                 try {
31                     Thread.sleep(2000);
32                 } catch (InterruptedException e) {
33                     e.printStackTrace();
34                 }finally{
35                     System.out.println("[1] done ");
36                     channel.basicAck(envelope.getDeliveryTag(), false);
37
38
39
40
41
42
43
44
45
}

```

Tasks Web Browser Console Servers Progress

Recv1 (3) [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86\_64\_1.6.0.u43\bin\javaw.exe (2018-1-28 下午8:11)

00:09:43

11.topic主题模式

```

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/topic/Send.java - MyEclipse Enterprise Workbench
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
Package Type Hierarchy Java Send.java
1 package com.mmr.rabbitmq.topic;
2
3 import java.io.IOException;
4 import java.util.concurrent.TimeoutException;
5
6 import com.mmr.rabbitmq.util.ConnectionUtils;
7 import com.rabbitmq.client.Channel;
8 import com.rabbitmq.client.Connection;
9
10 public class Send {
11
12
13     private static final String EXCHANGE_NAME = "test_exchange_topic";
14
15     public static void main(String[] args) throws IOException, TimeoutException {
16
17
18         Connection connection = ConnectionUtils.getConnection();
19
20         Channel channel = connection.createChannel();
21
22         //exchange
23         channel.exchangeDeclare(EXCHANGE_NAME, "topic");
24
25
26         String msgString="商品....";
27         channel.basicPublish(EXCHANGE_NAME, "goods.add", null, msgString.getBytes());
28         System.out.println("----send "+msgString);
29
30
31         channel.close();
32         connection.close();
33
34
35
36
}

```

Tasks Web Browser Console Servers Progress

Recv1 (3) [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86\_64\_1.6.0.u43\bin\javaw.exe (2018-1-28 下午8:11)

00:08:41

11.topic主题模式

模型。

```

graph LR
    P((P)) --> X((X))
    X -- "type=topic  
*.orange.*" --> Q1[Q1]
    X -- "*,*.rabit" --> Q2[Q2]
    Q1 --> C1((C1))
    Q2 -- "lazy#" --> C2((C2))
  
```

651 x 212

00:05:09

11.topic主题模式

Topic exchange  
将路由键和某模式匹配  
# 匹配一个或者多个  
\* 匹配一个

Topic Exchange

Messages:	routing key = usa.news	routing key = usa.weather	routing key = europe.news	routing key = europe.weather
Broker				
Exchange:				
Bindings:	binding key = usa.#	binding key = #.news	binding key = #.weather	binding key = europe.#
Queues:				

00:03:30

abbitmqMQ笔记.docx - Microsoft Word

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图

字体 段落 样式

导航

搜索文档

标题 | 页面 | 结果

- 1.简单队列的不足
- 2.Work queues工作队列
  - 2.1模型
  - 2.2生产者
  - 消费者1
  - 消费者2
  - 现象
- 2.公平分发fair diphach
  - 模型
  - 生产者
  - 消费者1
  - 消费者2
- 3.消息应答与消息持久化
  - 消息应答
  - 消息持久化
- 4.订阅模式publish/subscribe
  - 模型
  - 生产者
  - 消费者1
  - 消费者2
- 6. Exchange(交换机 转发器)
  - Fanout(不处理路由键)
  - Direct (处理路由键)
- 路由模式:
  - 模型
  - 生产者

第 22 页, 共 26 页 2709 个字 中文(中国)

**Broker**

The diagram illustrates a Broker architecture. At the top is an 'Exchange' represented by a red rectangle. Below it is a 'Bindings' section with three empty boxes labeled with asterisks (\*). At the bottom is a 'Queues' section with three boxes also labeled with asterisks (\*). A red arrow points from the 'Exchange' section down to the 'Bindings' section.

Topic exchange  
将路由键和某模式匹配

**Topic Exchange**

This diagram shows a 'Topic Exchange' setup. On the left, four messages are listed with their routing keys: 'routing key = usa.news', 'routing key = usa.weather', 'routing key = europe.news', and 'routing key = europe.weather'. These messages are shown entering a central 'Broker' box. Inside the broker, there are four 'binding key' boxes: 'binding key = usa.#', '#.news', '#.weather', and '#.europe'. Arrows show how specific messages are matched to these binding keys.

Messages: routing key = usa.news, routing key = usa.weather, routing key = europe.news, routing key = europe.weather

Broker

Bindings:

Queues: binding key = usa.#, #.news, #.weather, #.europe.#

rabbitmqMQ笔记.docx - Microsoft Word

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图

字体 段落 样式

导航

搜索文档

标题 | 页面 | 结果

- 1.3生产者生产消息
- 1.4消费者接受消息
- 1.5简单队列的不足
- 2.Work queues工作队列
  - 2.1模型
  - 2.2生产者
  - 消费者1
  - 消费者2
  - 现象
- 2.公平分发fair diphach
  - 模型
  - 生产者
  - 消费者1
  - 消费者2
- 3.消息应答与消息持久化
  - 消息应答
  - 消息持久化
- 4.订阅模式publish/subscribe
  - 模型
  - 生产者
  - 消费者1
  - 消费者2
- 6. Exchange(交换机 转发器)
  - Fanout(不处理路由键)
  - Direct (处理路由键)
- 路由模式:
  - 模型
  - 生产者

第 22 页, 共 25 页 2697 个字 中文(中国)

**路由模式**

**模型**

This diagram illustrates a Direct Exchange model. A producer node 'P' is connected to an exchange node 'X'. The exchange node has two output paths, each leading to a consumer node 'C1' and 'C2'. The top path is labeled with 'info' and 'amqp.gen-S9b...'. The bottom path is labeled with 'error', 'warning', and 'amqp.gen-Agl...'. The exchange node 'X' is labeled 'type=direct'.

P → X → C<sub>1</sub>, C<sub>2</sub>

**生产者**

```
public class Send {
    ...
    private static final String EXCHANGE_NAME="test_exchange_direct";
    ...
}
```

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/routing/Recv2.java - MyEclipse Enterprise Workbench

```

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
Package Type Hie Send.java Recv1.java Recv1.java Recv2.java
Other Projects
servlet
demo1
demo2
myrabbitmq
src/main/java
com.mmr.rabbitmq
ps
Recv1.java
Recv2.java
Send.java
routing
Recv1.java
Recv2.java
Send.java
simple
util
work
workfair
src/main/resources
src/test/java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml

Connection connection = ConnectionUtils.getConnection();
final Channel channel = connection.createChannel();

channel.queueDeclare(QUEUE_NAME, false, false, false, null);

channel.queueBind(QUEUE_NAME, EXCHANGE_NAME, "error");
channel.queueBind(QUEUE_NAME, EXCHANGE_NAME, "info");
channel.queueBind(QUEUE_NAME, EXCHANGE_NAME, "warning");

channel.basicQos(1);

//定义一个消费者
Consumer consumer=new DefaultConsumer(channel){
    //消息到达触发这个方法
    @Override
    public void handleDelivery(String consumerTag, Envelope envelope,
                               BasicProperties properties, byte[] body) throws IOException {
        String msg=new String(body,"utf-8");
        System.out.println("[1] Recv msg:"+msg);

        try {
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }finally{
            System.out.println("[2] done ");
            channel.basicAck(envelope.getDeliveryTag(), false);
        }
    }

    boolean autoAck=false;//自动应答 false
    channel.basicConsume(QUEUE_NAME,autoAck , consumer);
}
}

boolean autoAck=false;//自动应答 false
channel.basicConsume(QUEUE_NAME,autoAck , consumer);
}
}

```

Tasks Web Browser Console Servers Progress

No consoles to display at this time.

Writable Smart Insert 47 : 43

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/routing/Recv1.java - MyEclipse Enterprise Workbench

## 10.routing路由模式

```

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
Package Type Hie Recv1.java Recv1.java Recv1.java
Other Projects
servlet
demo1
demo2
myrabbitmq
src/main/java
com.mmr.rabbitmq
ps
Recv1.java
Recv2.java
Send.java
routing
Recv1.java
Send.java
simple
util
work
workfair
src/main/resources
src/test/java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml

public static void main(String[] args) throws IOException, TimeoutException {
    Connection connection = ConnectionUtils.getConnection();
    final Channel channel = connection.createChannel();

    channel.queueDeclare(QUEUE_NAME, false, false, false, null);

    channel.queueBind(QUEUE_NAME, EXCHANGE_NAME, "error");

    channel.basicQos(1);

    //定义一个消费者
    Consumer consumer=new DefaultConsumer(channel){
        //消息到达触发这个方法
        @Override
        public void handleDelivery(String consumerTag, Envelope envelope,
                                   BasicProperties properties, byte[] body) throws IOException {
            String msg=new String(body,"utf-8");
            System.out.println("[1] Recv msg:"+msg);

            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }finally{
                System.out.println("[2] done ");
                channel.basicAck(envelope.getDeliveryTag(), false);
            }
        }

        boolean autoAck=false;//自动应答 false
        channel.basicConsume(QUEUE_NAME,autoAck , consumer);
    }
}

boolean autoAck=false;//自动应答 false
channel.basicConsume(QUEUE_NAME,autoAck , consumer);
}
}

```

Tasks Web Browser Console Servers Progress

No consoles to display at this time.

Writable Smart Insert 54 : 9

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/routing/Send.java - MyEclipse Enterprise Workbench

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help

Package Type Hie

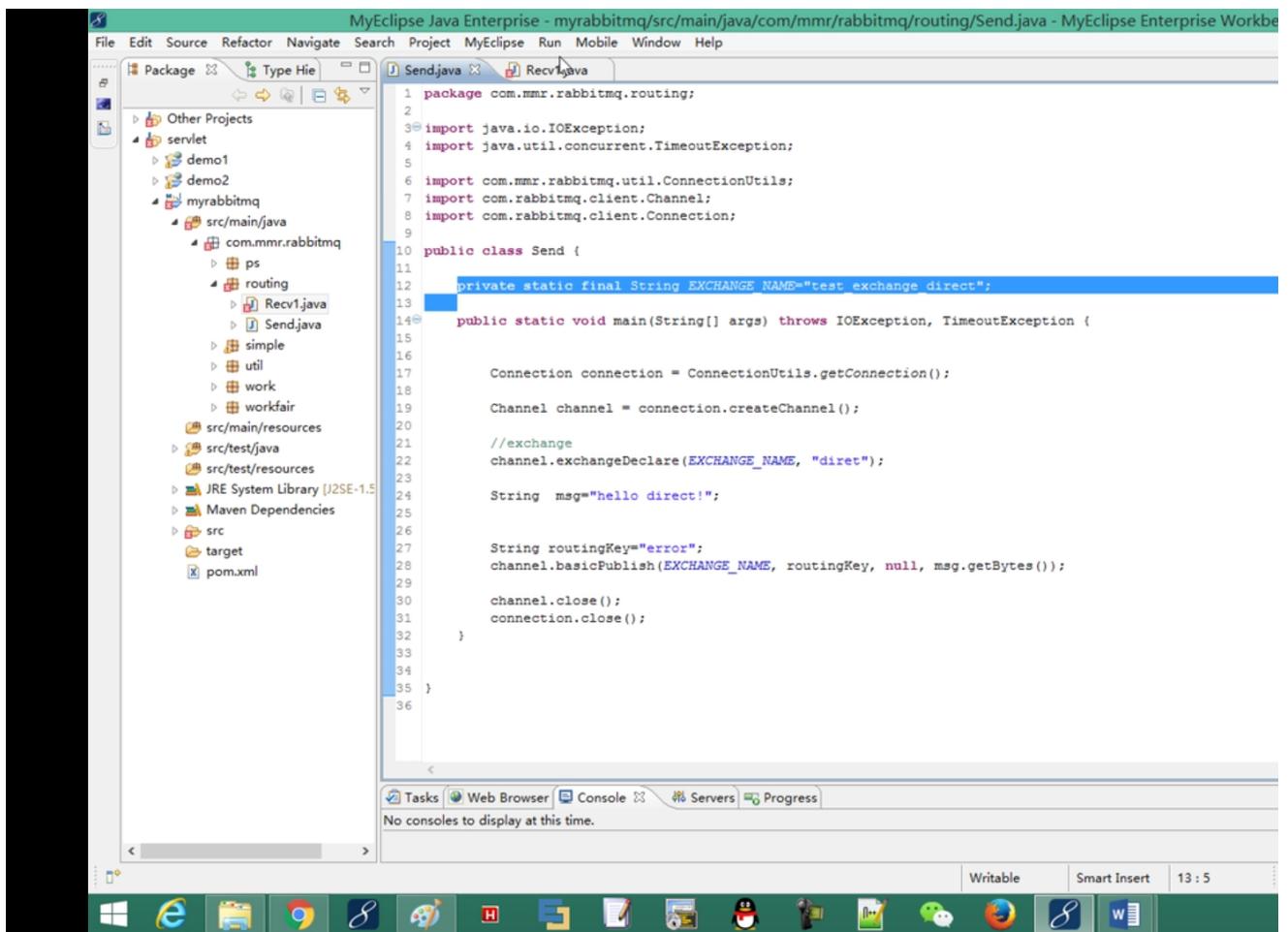
Sendjava Recv.java

```
1 package com.mmr.rabbitmq.routing;
2
3 import java.io.IOException;
4 import java.util.concurrent.TimeoutException;
5
6 import com.mmr.rabbitmq.util.ConnectionUtils;
7 import com.rabbitmq.client.Channel;
8 import com.rabbitmq.client.Connection;
9
10 public class Send {
11
12     private static final String EXCHANGE_NAME="test exchange direct";
13
14     public static void main(String[] args) throws IOException, TimeoutException {
15
16         Connection connection = ConnectionUtils.getConnection();
17
18         Channel channel = connection.createChannel();
19
20         //exchange
21         channel.exchangeDeclare(EXCHANGE_NAME, "direct");
22
23         String msg="hello direct!";
24
25         String routingKey="error";
26         channel.basicPublish(EXCHANGE_NAME, routingKey, null, msg.getBytes());
27
28         channel.close();
29         connection.close();
30
31     }
32
33
34
35 }
36
```

Tasks Web Browser Console Servers Progress

No consoles to display at this time.

Writable Smart Insert 13:5



手把手带你深入理解分布式缓存.pptx - WPS 演示

幻灯片放映 审阅 视图 云服务

新建幻灯片 版式 节

重置 文本框 形状 排列 轮廓 替换 选择窗格

28人在看

搜索框

手把手带你深入理解分布式缓存.pptx

## 一组比较数据

类别	耗时
内存访问	100ns
千兆网络发送 1MB 数据	10ms
从内存顺序读取 1MB 数据	0.25ms
机房内网络来回	0.5ms
异地机房之间网络来回	30~100ms
SATA 磁盘寻道	10ms
从SATA磁盘顺序读取 1MB 数据	20ms
固态盘SSD访问延迟	0.1~0.2ms

单击此处添加备注

A000120140530A21PPBG

备注

手把手带你深入理解分布式缓存.pptx - WPS 演示

幻灯片放映 审阅 视图 云服务

新建幻灯片 版式 节

重置 文本框 形状 排列 轮廓 替换 选择窗格

28人在看

搜索框

手把手带你深入理解分布式缓存.pptx

## 一组比较数据

类别	耗时
内存访问	100ns
千兆网络发送 1MB 数据	10ms
从内存顺序读取 1MB 数据	0.25ms
机房内网络来回	0.5ms
异地机房之间网络来回	30~100ms
SATA 磁盘寻道	10ms
从SATA磁盘顺序读取 1MB 数据	20ms
固态盘SSD访问延迟	0.1~0.2ms

单击此处添加备注

A000120140530A21PPBG

备注

4 倍

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/ps/Recv1.java - MyEclipse Enterprise Workbench

## 09.publish\_subscribe订阅模式

```
private static final String QUEUE_NAME="test_queue_fanout_email";
private static final String EXCHANGE_NAME="test_exchange_fanout";
public static void main(String[] args) throws IOException, TimeoutException {
    Connection connection = ConnectionUtils.getConnection();
    final Channel channel = connection.createChannel();

    //队列声明
    channel.queueDeclare(QUEUE_NAME, false, false, false, null);

    //绑定队列到交换机 转发器
    channel.queueBind(QUEUE_NAME, EXCHANGE_NAME, "");

    channel.basicQos(1);//保证一次只分发一个

    //定义一个消费者
    Consumer consumer=new DefaultConsumer(channel){
        //消息到达 触发这个方法
        @Override
        public void handleDelivery(String consumerTag, Envelope envelope,
                                   BasicProperties properties, byte[] body) throws IOException {
            String msg=new String(body,"utf-8");
            System.out.println("[1] Recv msg:"+msg);

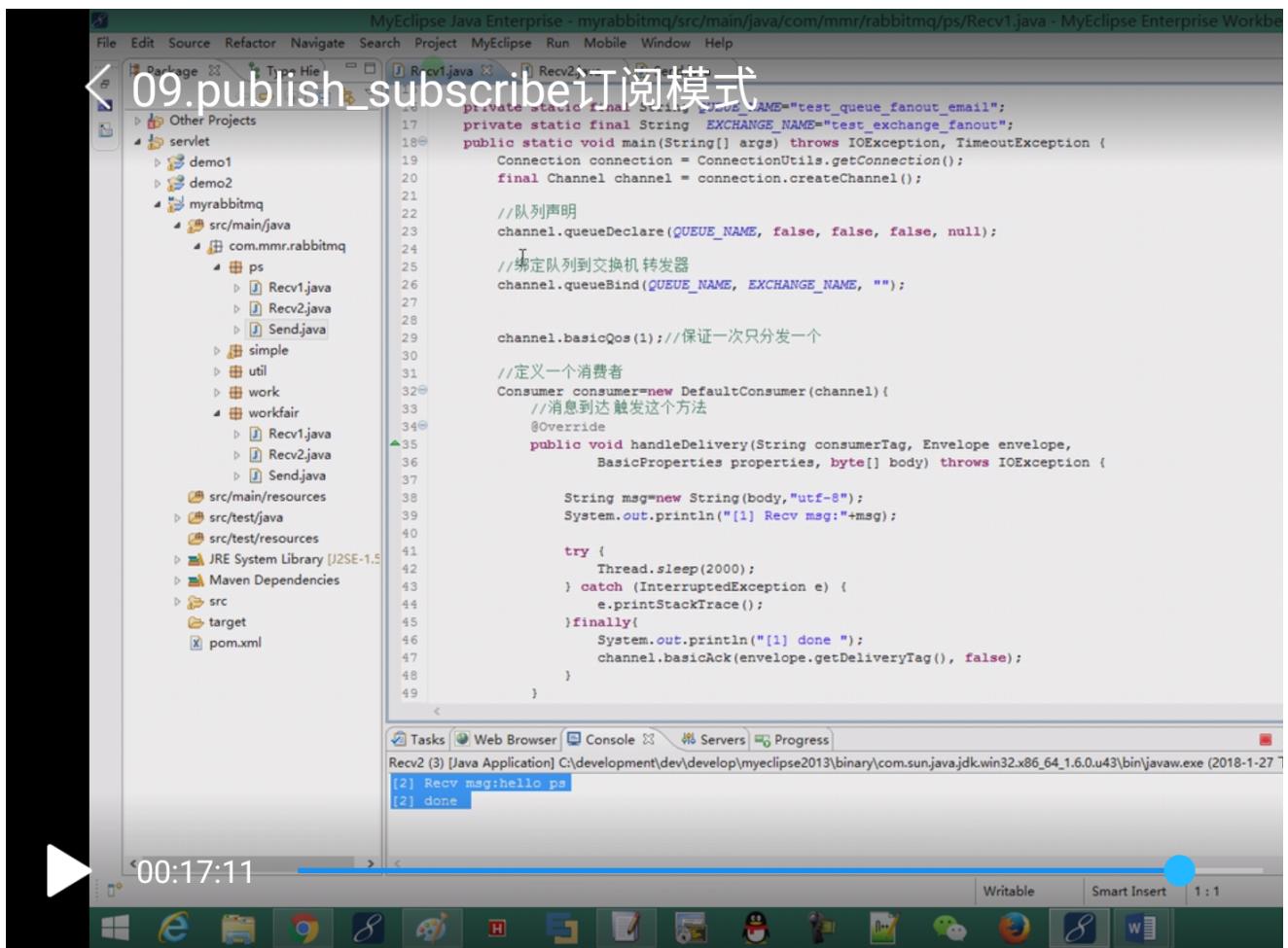
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }finally{
                System.out.println("[1] done ");
                channel.basicAck(envelope.getDeliveryTag(), false);
            }
        }
    };
    channel.basicConsume(QUEUE_NAME, true, consumer);
}
```

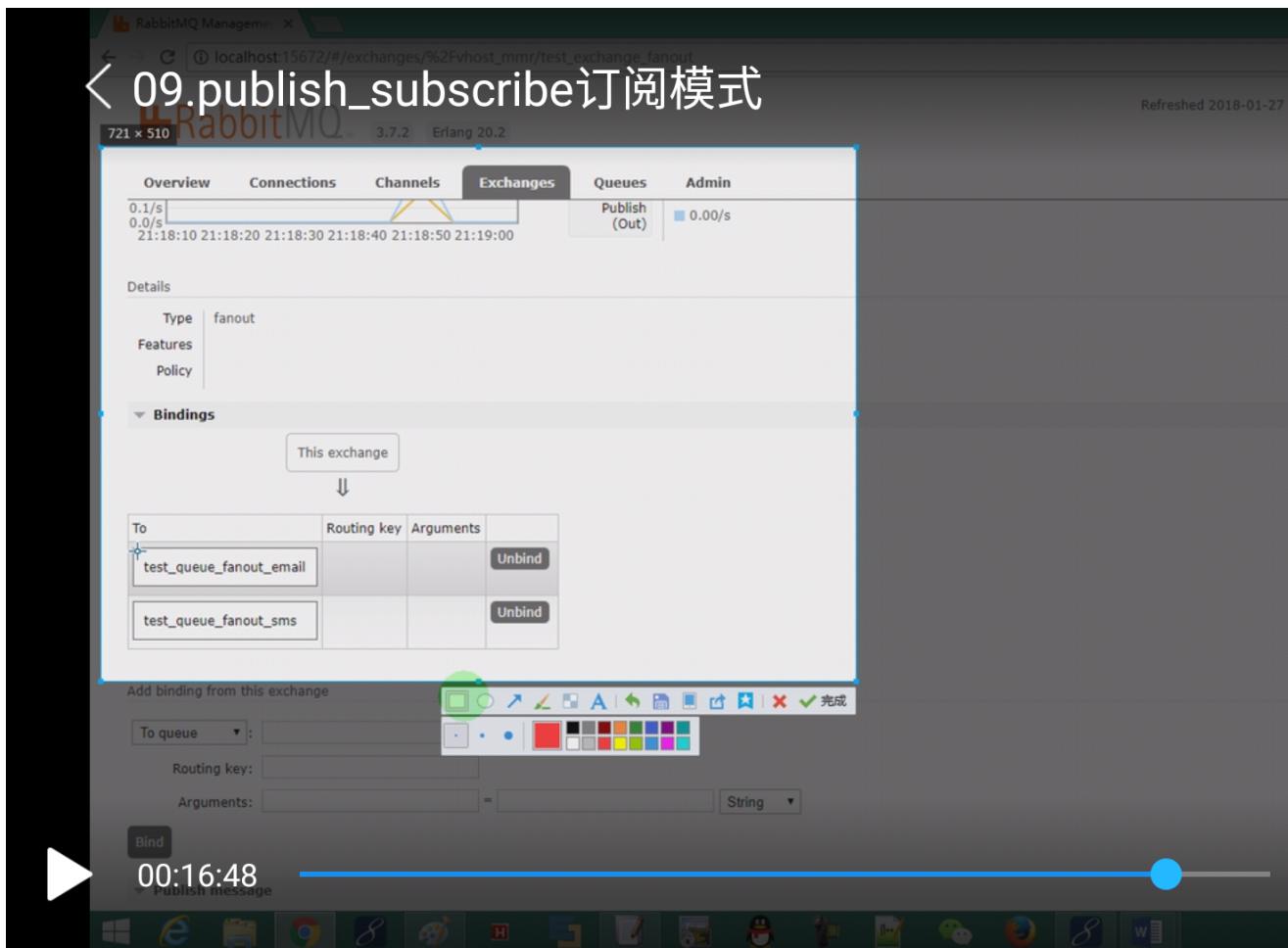
Tasks Web Browser Console Servers Progress

Recv2 (3) [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86\_64\_1.6.0.u43\bin\javaw.exe (2018-1-27)

[2] Recv msg:hello ps  
[2] done

00:17:11





MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/ps/Send.java - MyEclipse Enterprise Workbench

```

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
Package Type Hie
Other Projects
  servlet
    demo1
    demo2
  myrabbitmq
    src/main/java
      com.mmr.rabbitmq
        ps
          Send.java
        simple
        util
        work
        workfair
      src/main/resources
    src/test/java
    src/test/resources
  JRE System Library [J2SE-1.5]
  Maven Dependencies
  src
  target
  pom.xml

Send.java
1 package com.mmr.rabbitmq.ps;
2
3 import java.io.IOException;
4 import java.util.concurrent.TimeoutException;
5
6 import com.mmr.rabbitmq.util.ConnectionUtils;
7 import com.rabbitmq.client.Channel;
8 import com.rabbitmq.client.Connection;
9
10 public class Send {
11
12     private static final String EXCHANGE_NAME="test_exchange_fanout";
13     public static void main(String[] args) throws IOException, TimeoutException {
14
15         Connection connection = ConnectionUtils.getConnection();
16
17         Channel channel = connection.createChannel();
18
19         //声明交换机
20         channel.exchangeDeclare(EXCHANGE_NAME, "fanout");//分发
21
22         //发送消息
23         String msg="hello ps";
24
25         channel.basicPublish(EXCHANGE_NAME, "", null, msg.getBytes());
26
27         System.out.println("Send :" +msg);
28
29         channel.close();
30         connection.close();
31     }
32
33
34

```

Tasks Web Browser Console Servers Progress

<terminated> Send (4) [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86\_64\_1.6.0.u43\bin\javaw.exe

Send :hello ps

Writable Smart Insert 30 : 25

rabbitmqMQ笔记.docx - Microsoft Word

文件 开始 插入 设计 页面布局 引用 邮件 审阅 视图

表格工具 设计 布局

第一列 汇总 第二列 第三列 第四列 第五列 第六列 第七列 第八列 第九列

表格样式选项

边框 边框刷

0.5磅

笔颜色

边框

# 09.publish\_subscribe订阅模式

导航

搜索文档

标题 | 页面 | 结果

用户界面  
virtual hosts管理  
2.java操作队列  
  1. 简单队列  
    1.1模型  
    1.2获取MQ连接  
    1.3生产者生产消息  
    1.4消费者接受消息  
    1.5简单队列的不足  
  2.Work queues工作队列  
    2.1模型  
    2.2生产者  
    消费者1  
    消费者2  
    现象  
  2.公平分发fair dispatch  
    模型  
    生产者  
    消费者1  
    消费者2  
  3.消息应答与消息持久化  
    消息应答  
    消息的持久化  
  订阅模式  
    模型

生产者 00:06:32

第 17 页, 共 17 页 1937 个字 中文(中国)

解读:

1. 一个生产者,多个消费者
2. 每一个消费者都有自己的队列
3. 生产者没有直接把消息发送到队列 而是发到了交换机 转发器 exchange
4. 每个队列都要绑定到交换机上
5. 生产者发送的消息 经过交换机 到达队列 就能实现 一个消息被多个消费者消费

注册 →邮件→短信→

生产者

```
graph LR; P((P)) --> X((X)); X --> C1((C1)); X --> C2((C2)); X --> C3((C3))
```

Windows taskbar icons: File Explorer, Edge, File Manager, Google Chrome, Paint, Task View, File History, Control Panel, File Explorer, File History, File Explorer, File History.

< 09 publish\_subscribe订阅模式

订阅模式

模型

```

graph LR
    P((P)) --> X((X))
    X --> C1((C1))
    X --> C2((C2))
  
```

00:02:41

第 17 页, 共 17 页 1821 个字 英语(美国)

< 08 消息应答ack与消息持久化durable

`boolean autoAck=false;` (手动模式), 如果有一个消费者挂掉,  
就会交付给其他消费者,rabbitmq 支持消息应答,消费者发送一个消息应答,高数 rabbitmq  
个消息我已经处理完成 你可以删了,然后 rabbitmq 就删除内存中的消息。

消息应答默认是打开的, `false`

Message acknowledgment:  
大家想想如果我 rabbitmq 挂了 我们的消息任然会丢失!!!!

• 消息的持久化

```

//声明队列
boolean durable= false;
channel.queueDeclare(QUEUE_NAME, durable, false, false, null)
  
```

我们将程序中的`boolean durable= false;`改成`true`;是不可以的,尽管代码是正确的,他也不会运行成功!因为我们已经定义了一个叫 `test_work_queue` 这个queue;持久化,rabbitmq不准许重新定义(不同参数)一个已存在的队列

00:16:31

第 16 页, 共 16 页 1/1811 个字 英语(美国)

abbitmqMQ笔记.docx - Microsoft Word

开始 插入 设计 页面布局 引用 邮件 审阅 视图

Courier New - 10 A Aa Aa w A

B I abc x, x<sup>2</sup> A Aa Aa w A

AaBbCcDd AaBbCcDd AaBbCcDd AaBbCcDd

段落 样式

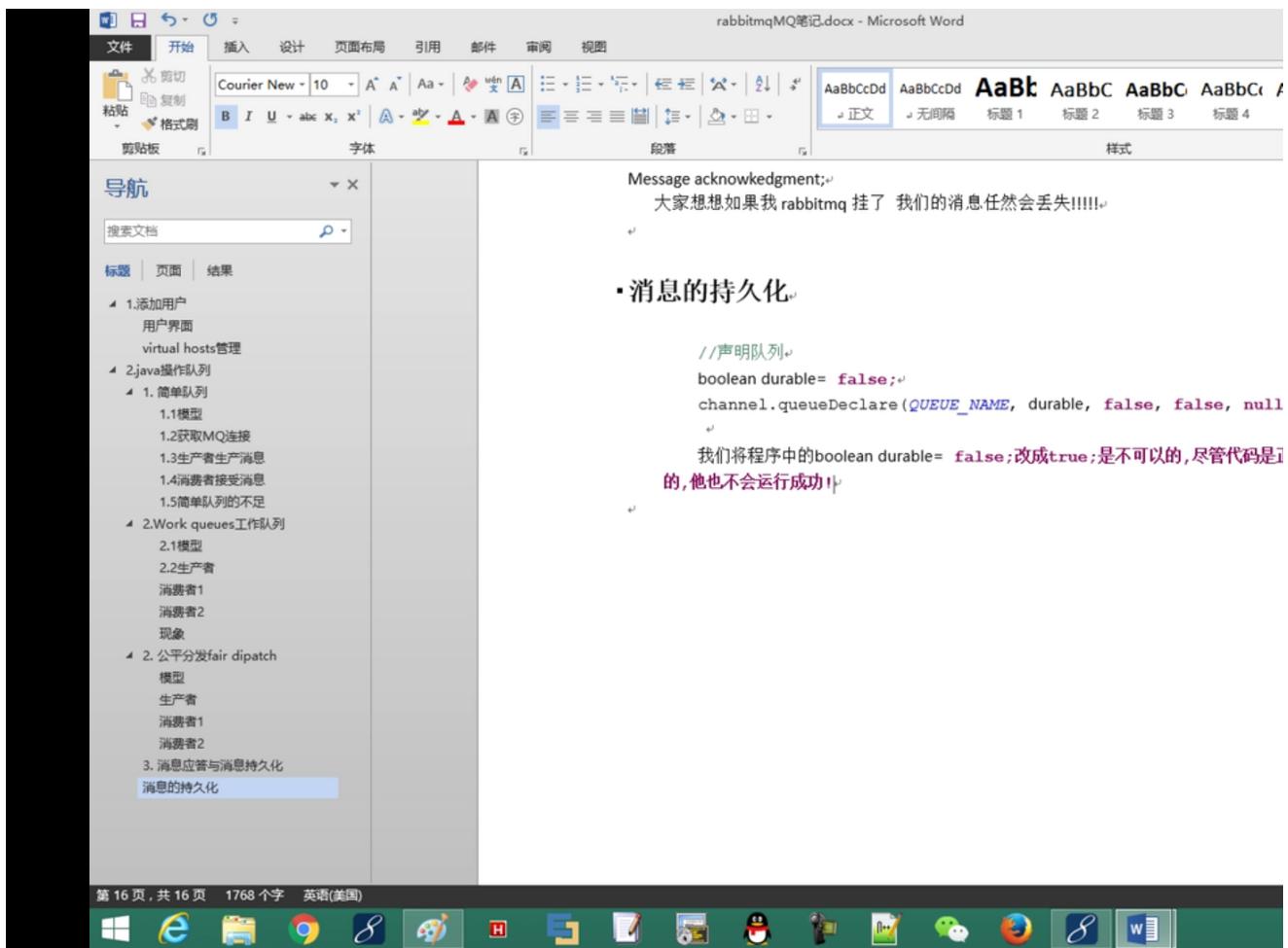
Message acknowledgment;  
大家想想如果我 rabbitmq 挂了 我们的消息任然会丢失!!!!!

• 消息的持久化

```
//声明队列
boolean durable= false;
channel.queueDeclare(QUEUE_NAME, durable, false, false, null)
```

我们将程序中的boolean durable= false;改成true;是不可以的,尽管代码是正确的,他也不会运行成功!|

第 16 页 , 共 16 页 1768 个字 英语(美国)



08.消息应答ack与消息持久化durable

•3. 消息应答与消息持久化

```
boolean autoAck=false;+  
channel.basicConsume(QUEUE_NAME,autoAck , consumer);+  
  
boolean autoAck=true; (自动确认模式)一旦rabbitmq将消息分发给消费者,就会从中删除;+  
这种情况下,如果杀死正在执行的消费者,就会丢失正在处理的消息;+  
  
boolean autoAck=false; (手动模式),如果有消费者挂掉,+  
就会交付给其他消费者,rabbitmq 支持消息应答,消费者发送一个消息应答,高数 rabbitmq  
个消息我已经处理完成 你可以删了,然后 rabbitmq 就删除内存中的消息;+
```

00:07:42 第 16 页, 共 16 页 1677 个字 英语(美国)

08.消息应答ack与消息持久化durable

•3. 消息应答与消息持久化

```
boolean autoAck=false;+  
channel.basicConsume(QUEUE_NAME,autoAck , consumer);+  
  
boolean autoAck=true; (自动确认模式);+  
  
boolean autoAck=false; (手动模式),如果有消费者挂掉,+  
就会交付给其他消费者,rabbitmq 支持消息应答,消费者发送一个消息应答,高数 rabbitmq  
个消息我已经处理完成 你可以删了,然后 rabbitmq 就删除内存中的消息;+
```

00:02:17 第 16 页, 共 16 页 8/1543 个字 英语(美国)

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/workfair/Recv2.java - MyEclipse Enterprise Work

## 07.work queues工作队列之Fair dispatch

```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
Package Explorer Type Hierarchy
Other Projects
  servlet
    demo1
    demo2
  myrabbitmq
    src/main/java
      com.mmr.rabbitmq
        simple
        util
        work
          Recv1.java
          Recv2.java
          Send.java
        workfair
          Recv1.java
          Recv2.java
          Send.java
    src/main/resources
    src/test/java
    src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml

18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

public static void main(String[] args) throws Exception, TimeoutException {
    //获取连接
    Connection connection = ConnectionUtils.getConnection();
    //获取channel
    Channel channel = connection.createChannel();
    //声明队列
    channel.queueDeclare(QUEUE_NAME, false, false, false, null);
    channel.basicQos(1); //保证一次只分发一个

    //定义一个消费者
    Consumer consumer=new DefaultConsumer(channel){
        //消息到达触发这个方法
        @Override
        public void handleDelivery(String consumerTag, Envelope envelope,
                                   BasicProperties properties, byte[] body) throws IOException {
            String msg=new String(body,"utf-8");
            System.out.println("[2] Recv msg:"+msg);

            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }finally{
                System.out.println("[2] done ");
            }
        };
        boolean autoAck=false;
        channel.basicConsume(QUEUE_NAME,autoAck , consumer);
    }
}

00:08:37
Tasks Web Browser Console Servers Progress
No consoles to display at this time.
Writable Smart Insert 43 : 53

```

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/workfair/Recv1.java - MyEclipse Enterprise Work

```

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
< 07.work queues工作队列之Fair dispatch
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

18     //获取连接
19     Connection connection = ConnectionUtils.getConnection();
20     //获取channel
21     final Channel channel = connection.createChannel();
22     //声明队列
23     channel.queueDeclare(QUEUE_NAME, false, false, false, null);
24
25     channel.basicQos(1); //保证一次只分发一个
26
27     //定义一个消费者
28     Consumer consumer=new DefaultConsumer(channel){
29         //消息到达触发这个方法
30         @Override
31         public void handleDelivery(String consumerTag, Envelope envelope,
32             BasicProperties properties, byte[] body) throws IOException {
33
34             String msg=new String(body,"utf-8");
35             System.out.println("[1] Recv msg:"+msg);
36
37             try {
38                 Thread.sleep(2000);
39             } catch (InterruptedException e) {
40                 e.printStackTrace();
41             }finally{
42                 System.out.println("[1] done ");
43                 channel.basicAck(envelope.getDeliveryTag(), false);
44             }
45
46         };
47
48         boolean autoAck=true;//自动应答
49         channel.basicConsume(QUEUE_NAME,autoAck , consumer);
50
51     }
52
53 }

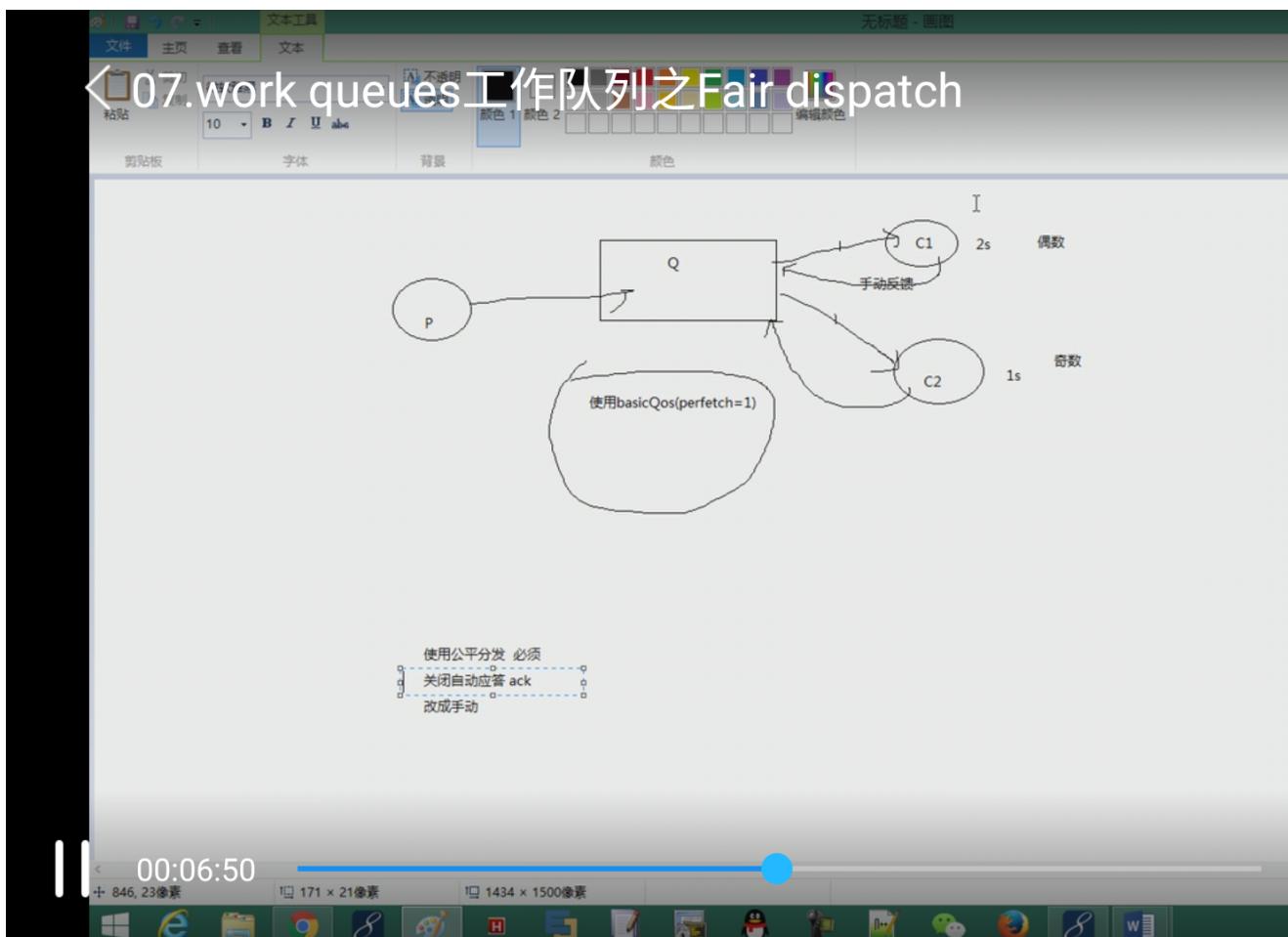
```

Tasks Web Browser Console Servers Progress

No consoles to display at this time.

00:07:52

Writable Smart Insert 49 : 32



MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/workfair/Send.java - MyEclipse Enterprise Workbench

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help

Package Type Hie

Send.java Recv1.java Send.java

```
private static final String QUEUE_NAME="test_work_queue";

public static void main(String[] args) throws IOException, TimeoutException, InterruptedException {
    //获取连接
    Connection connection = ConnectionUtils.getConnection();

    //获取channel
    Channel channel = connection.createChannel();

    //声明队列
    channel.queueDeclare(QUEUE_NAME, false, false, false, null);

    /**
     * 每个消费者 发送确认消息之前,消息队列不发送下一个消息到消费者,一次只处理一个消息
     *
     * 限制发送给同一个消费者 不得超过一条消息
     */
    int prefetchCount=1;
    channel.basicQos(prefetchCount);

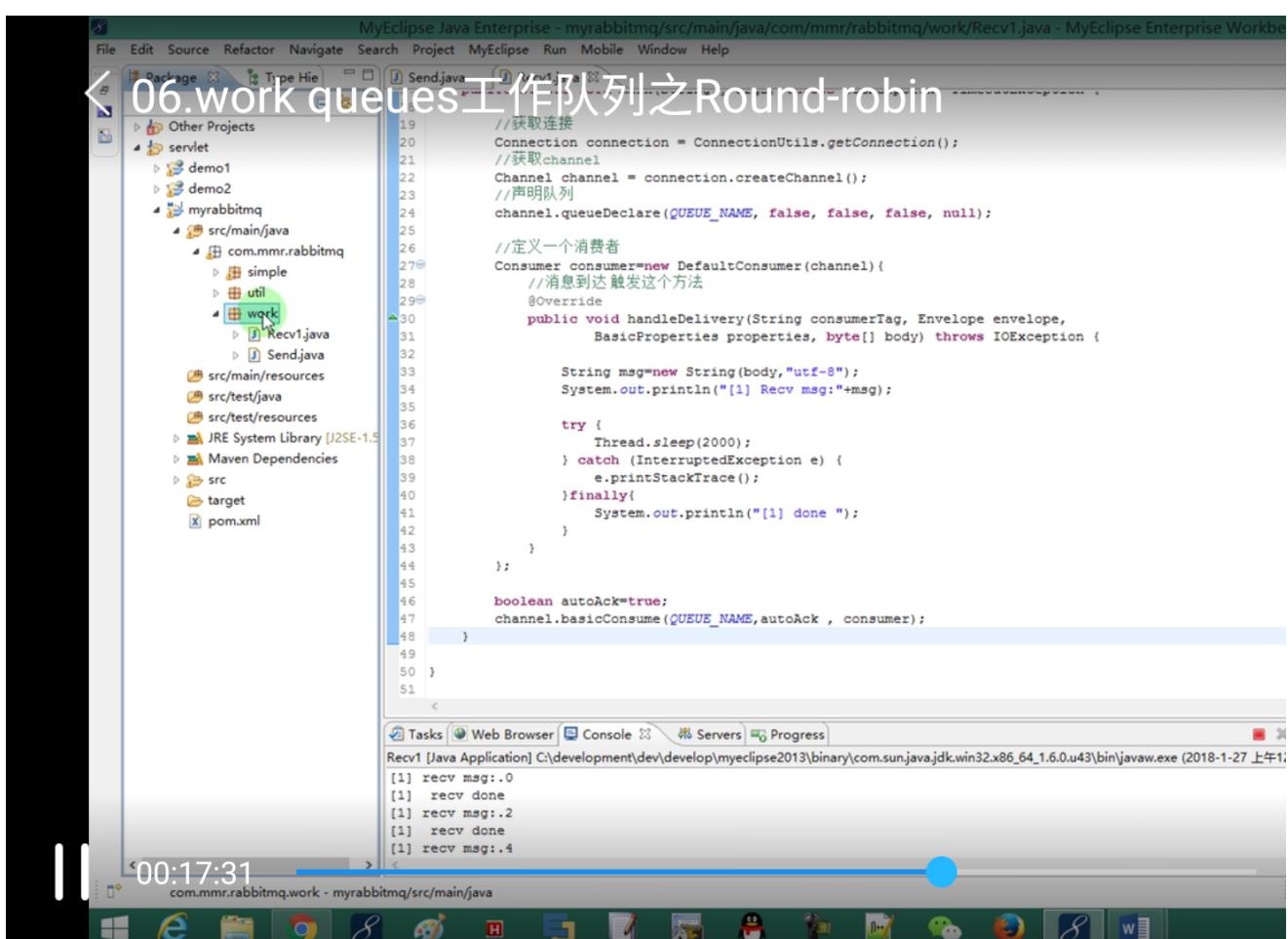
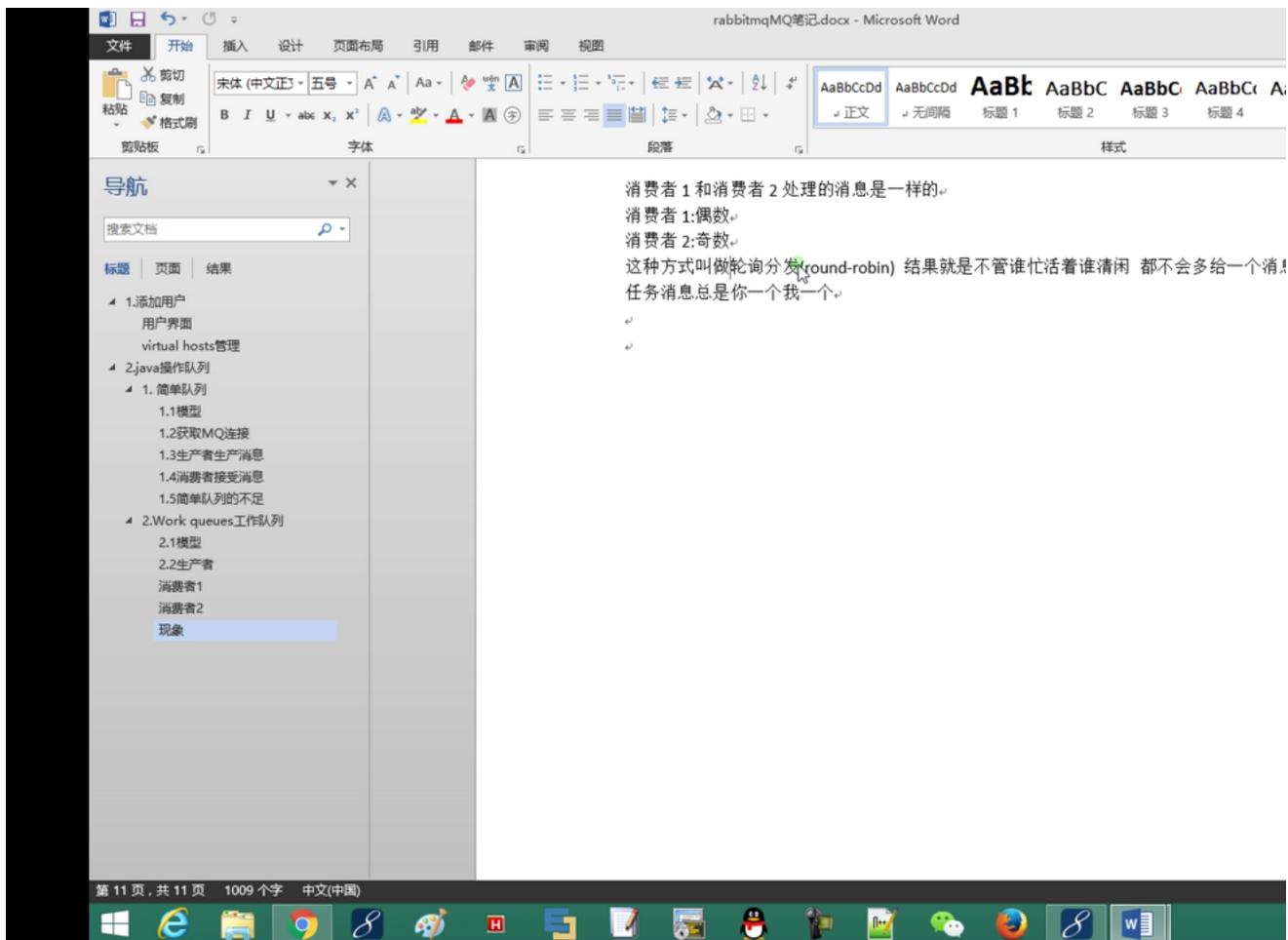
    for (int i = 0; i <50; i++) {
```

Tasks Web Browser Console Servers Progress

No consoles to display at this time.

Writable Smart Insert 37 : 9





MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/work/Recv1.java - MyEclipse Enterprise Workbench

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help

Package Type Hie

Other Projects  
servlet  
demo1  
demo2  
myrabbitmq  
src/main/java  
com.mmr.rabbitmq  
simple  
util  
work  
Recv1.java  
Send.java  
src/main/resources  
src/test/java  
src/test/resources  
JRE System Library [J2SE-1.5]  
Maven Dependencies  
src  
target  
pom.xml

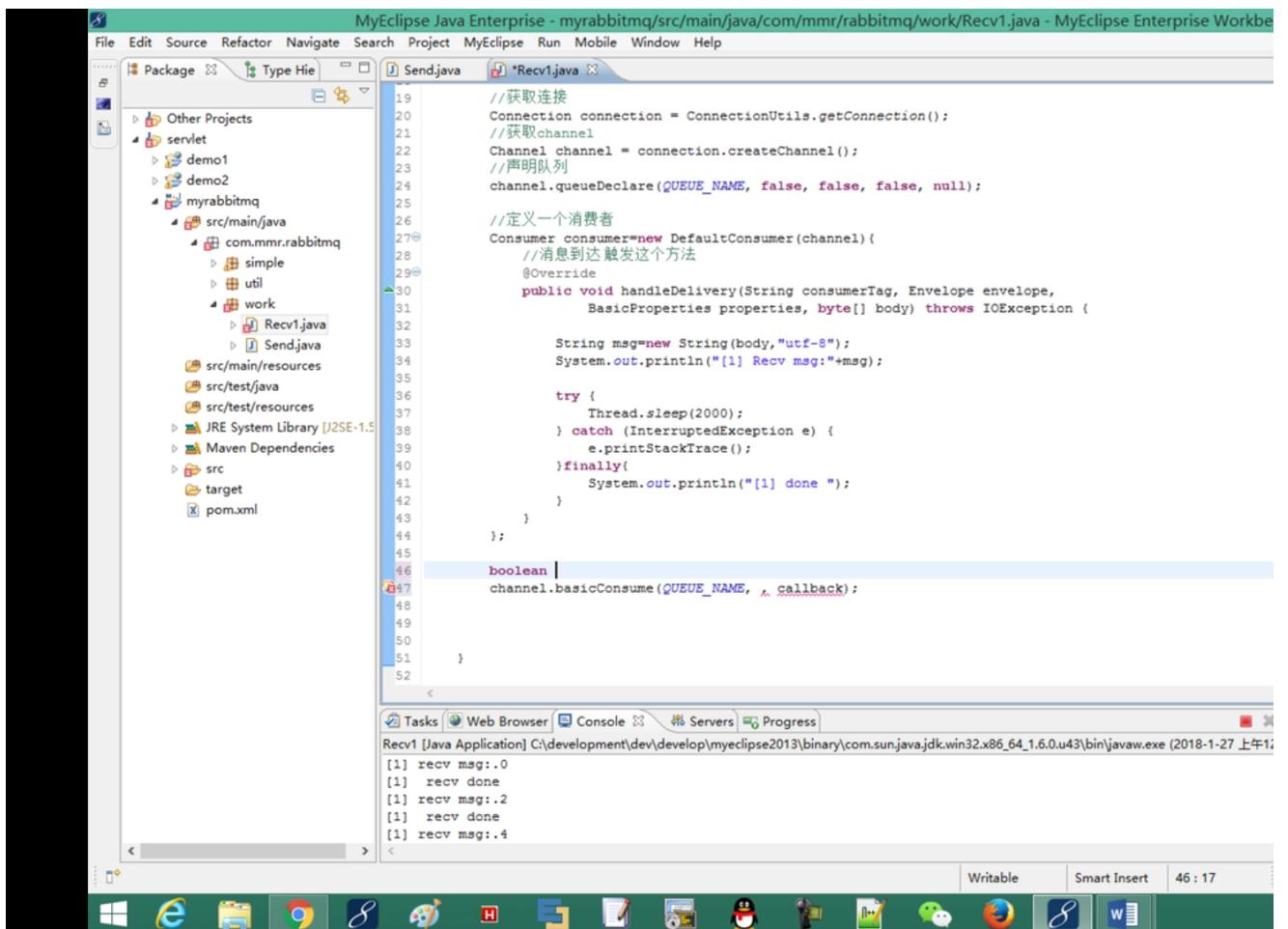
Send.java Recv1.java

```
19 //获取连接
20 Connection connection = ConnectionUtils.getConnection();
21 //获取channel
22 Channel channel = connection.createChannel();
23 //声明队列
24 channel.queueDeclare(QUEUE_NAME, false, false, false, null);
25
26 //定义一个消费者
27 Consumer consumer=new DefaultConsumer(channel){
28     //消息到达触发这个方法
29     @Override
30     public void handleDelivery(String consumerTag, Envelope envelope,
31         BasicProperties properties, byte[] body) throws IOException {
32
33         String msg=new String(body,"utf-8");
34         System.out.println("[1] Recv msg:"+msg);
35
36         try {
37             Thread.sleep(2000);
38         } catch (InterruptedException e) {
39             e.printStackTrace();
40         }finally{
41             System.out.println("[1] done ");
42         }
43     }
44
45
46 boolean |
47 channel.basicConsume(QUEUE_NAME, _callback);
48
49
50
51
52
```

Tasks Web Browser Console Servers Progress

Recv1 [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86\_64\_1.6.0.u43\bin\javaw.exe (2018-1-27 上午11:11:45)
[1] recv msg::0
[1] recv done
[1] recv msg::2
[1] recv done
[1] recv msg::4

Writable Smart Insert 46 : 17



MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/work/Send.java - MyEclipse Enterprise Workbench

```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
Package Type Hie
Other Projects
  servlet
    demo1
    demo2
  myrabbitmq
    src/main/java
      com.mmr.rabbitmq
        simple
        util
        work
          Send.java
    src/main/resources
    src/test/java
    src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml

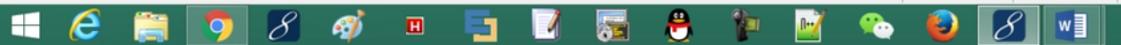
Send.java
23
24   //获取channel
25   Channel channel = connection.createChannel();
26
27   //声明队列
28   channel.queueDeclare(QUEUE_NAME, false, false, false, null);
29
30   for (int i = 0; i <50; i++) {
31
32     String msg="hello "+i;
33
34     channel.basicPublish("", QUEUE_NAME, null, msg.getBytes());
35     Thread.sleep(i*20);
36   }
37
38   channel.close();
39   connection.close();
40
41 }
42
43
44
45
```

Tasks Web Browser Console Servers Progress

Recv1 [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86\_64\_1.6.0.u43\bin\javaw.exe (2018-1-27 上午11:11:11)

```
[1] recv msg:.0
[1] recv done
[1] recv msg:.2
[1] recv done
[1] recv msg:.4
[1] recv done
[1] recv msg:.6
[1] recv done
[1] recv msg:.8
[1] recv done
[1] recv msg:.10
[1] recv done
[1] recv msg:.12
[1] recv done
[1] recv msg:.14
[1] recv done
[1] recv msg:.16
<
```

Writable Smart Insert 40 : 9



MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/work/Send.java - MyEclipse Enterprise Workbench

```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
Package Type Hie
Other Projects
  servlet
    demo1
    demo2
  myrabbitmq
    src/main/java
      com.mmr.rabbitmq
        simple
        util
        work
          Send.java
    src/main/resources
    src/test/java
    src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml

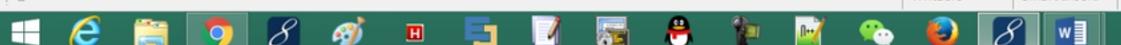
Send.java
17
18   private static final String QUEUE_NAME="test_work_queue";
19
20   public static void main(String[] args) throws IOException, TimeoutException {
21     //获取连接
22     Connection connection = ConnectionUtils.getConnection();
23
24     //获取channel
25     Channel channel = connection.createChannel();
26
27     //声明队列
28     channel.queueDeclare(QUEUE_NAME, false, false, false, null);
29
30     for (int i = 0; i <50; i++) {
31
32       String msg="hello "+i;
33
34       channel.basicPublish("", routingKey, props, body);
35
36     }
37
38
39 }
```

Tasks Web Browser Console Servers Progress

Recv1 [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86\_64\_1.6.0.u43\bin\javaw.exe (2018-1-27 上午11:11:11)

```
[1] recv msg:.0
[1] recv done
[1] recv msg:.2
[1] recv done
[1] recv msg:.4
[1] recv done
[1] recv msg:.6
[1] recv done
[1] recv msg:.8
[1] recv done
[1] recv msg:.10
[1] recv done
[1] recv msg:.12
[1] recv done
[1] recv msg:.14
[1] recv done
[1] recv msg:.16
<
```

Writable Smart Insert 34 : 63



rabbitmqMQ笔记.docx - Microsoft Word

开始 插入 设计 页面布局 引用 邮件 审阅 视图

字体 颜色 样式

导航

搜索文档

标题 | 页面 | 结果

▲ 1.添加用户  
    用户界面  
    virtual hosts管理

▲ 2.java操作队列  
    ▲ 1.简单队列  
        1.1模型  
        1.2获取MQ连接  
        1.3生产者生产消息  
        1.4消费者接受消息  
        1.5简单队列的不足

▲ Work queues工作队列  
    模型

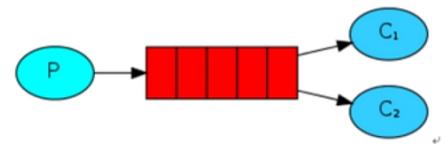
• 1.5 简单队列的不足。  
耦合性高,生产者一一对应消费者(如果我想有多个消费者消费队列中消息,这时候就不行)  
队列名变更 这时候得同时变更。

• Work queues 工作队列

• 模型。

第7页,共7页 519个字 中文(中国)

Windows图标栏: Internet Explorer, File Explorer, Google Chrome, Edge, Microsoft Paint, Notepad, Task View, File Explorer, File Explorer.



06.work queues工作队列之Round robin

耦合性高,生产者一一对应消费者(如果我想有多个消费者消费队列中消息,这时候就不行)  
队列名变更 这时候得同时变更。

**Work queues 工作队列**

```

graph LR
    P((P)) --> Queue[Queue]
    Queue --> C1((C1))
    Queue --> C2((C2))

```

00:02:30 第 7 页, 共 7 页 519 个字 英语(美国)

05.Java操作simple简单队列新api

```

public void handleDelivery(String consumerTag, Envelope envelope,
                           BasicProperties properties, byte[] body) throws IOException {
    String msg=new String(body,"utf-8");
    System.out.println("new api recv:"+msg);
}
    channel.basicConsume(QUEUE_NAME, true,consumer);
}

private static void oldapi() throws IOException, TimeoutException {
    Connection connection = ConnectionUtil.getConnection();
    Channel channel = connection.createChannel();
    channel.queueDeclare(QUEUE_NAME, false, false, false, null);
    DefaultConsumer consumer = new DefaultConsumer(channel) {
        @Override
        public void handleDelivery(String consumerTag, Envelope envelope,
                                   BasicProperties properties, byte[] body) throws IOException {
            String msg=new String(body,"utf-8");
            System.out.println("old api recv:"+msg);
        }
    };
    channel.basicConsume(QUEUE_NAME, true,consumer);
}

```

00:08:16

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/simple/Recv.java - MyEclipse Enterprise Workbench

```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
```

Package Type Hie

Other Projects

- servlet
  - demo1
  - demo2
- myrabbitmq
  - src/main/java
    - com.mmr.rabbitmq
      - simple
        - Recvjava
        - Sendjava
      - util
    - ConnectionUtils.java
  - src/main/resources
  - src/test/java
  - src/test/resources
- JRE System Library [J2SE-1.5]
- Maven Dependencies
- src
- target
- pom.xml

Recv.java

```
33     Connection connection = ConnectionUtils.getConnection();
34     // 创建频道
35     Channel channel = connection.createChannel();
36
37     //队列声明
38     channel.queueDeclare(QUEUE_NAME, false, false, false, null);
39
40     DefaultConsumer consumer = new DefaultConsumer(channel) {
41         @Override
42         public void handleDelivery(String consumerTag, Envelope envelope,
43             BasicProperties properties, byte[] body) throws IOException {
44
45             String msg=new String(body,"utf-8");
46             System.out.println("new api recv:"+msg);
47         }
48     };
49
50     //监听队列 android
51     channel.basicConsume(QUEUE_NAME, true,consumer);
52 }
53
54
55     private static void oldapi() throws IOException, TimeoutException,
56                                         InterruptedException {
57         // 获取连接
58         Connection connection = ConnectionUtils.getConnection();
59
60         // 创建频道
61         Channel channel = connection.createChannel();
62         // 定义队列的消费者
63         QueueingConsumer consumer = new QueueingConsumer(channel);
64         // 监听队列
```

Tasks Web Browser Console Servers Progress

Recv [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86\_64\_1.6.0.u43\bin\javaw.exe (2018-1-26 下午11:35:34)

```
[recv] msg:hello simple !
[recv] msg:hello simple !
[recv] msg:hello simple !
```

Writable Smart Insert 52 : 1

The screenshot shows the MyEclipse Java Enterprise interface. The left pane displays the project structure under 'Other Projects' and 'myrabbitmq'. The right pane shows the 'Recv.java' file with Java code for connecting to a RabbitMQ queue and handling messages. Below the code editor is a 'Console' tab showing the output of the application's execution, which includes three lines of text: '[recv] msg:hello simple !'. The bottom of the screen features a Windows taskbar with various icons.

< 05. Java操作simple简单队列新api

```

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/simple/Recv.java - MyEclipse Enterprise Workbench
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
Package ConnectionUtils.java Send.java Recv.java
src/main/java com.mmr.rabbitmq.simple Recv.java
src/main/java com.mmr.rabbitmq.simple ConnectionUtils.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml

private static final String QUEUE_NAME = "test_simple_queue";
@SuppressWarnings("deprecation")
public static void main(String[] args) throws IOException,
TimeoutException, ShutdownSignalException,
ConsumerCancelledException, InterruptedException {
    // 获取连接
    Connection connection = ConnectionUtils.getConnection();
    // 创建频道
    Channel channel = connection.createChannel();
    //队列声明
    channel.queueDeclare(QUEUE_NAME, false, false, false, null);
    DefaultConsumer consumer = new DefaultConsumer(channel) {
        @Override
        public void handleDelivery(String consumerTag, Envelope envelope,
                                   BasicProperties properties, byte[] body) throws IOException {
            String msg=new String(body,"utf-8");
            System.out.println("new api recv:"+msg);
        }
    };
    //监听队列
    channel.basicConsume(QUEUE_NAME, true, consumer);
}
private static void main(String[] args) throws IOException,
InterruptedException {
    Connection connection = ConnectionUtils.getConnection();
    // 创建频道
    Channel channel = connection.createChannel();
    // 定义队列的消费者
    QueueingConsumer consumer = new QueueingConsumer(channel);
    // 监听队列
}

Tasks Web Browser Console Servers Progress
Recv [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86_64_1.6.0.u43\bin\javaw.exe (2018-1-26 下午11:35:34)
[recv] msg:hello simple !
[recv] msg:hello simple !
[recv] msg:hello simple !
[recv] msg:hello simple !

```

00:06:05

```

MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/simple/Recv.java - MyEclipse Enterprise Workbench
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
Package ConnectionUtils.java Send.java Recv.java
src/main/java com.mmr.rabbitmq.simple Recv.java
src/main/java com.mmr.rabbitmq.simple ConnectionUtils.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml

private static final String QUEUE_NAME = "test_simple_queue";
@SuppressWarnings("deprecation")
public static void main(String[] args) throws IOException,
TimeoutException, ShutdownSignalException,
ConsumerCancelledException, InterruptedException {
    // 获取连接
    Connection connection = ConnectionUtils.getConnection();
    // 创建频道
    Channel channel = connection.createChannel();
    //队列声明
    channel.queueDeclare(QUEUE_NAME, false, false, false, null);
    new DefaultConsumer(channel) {
        @Override
        public void handleDelivery(String consumerTag, Envelope envelope,
                                   BasicProperties properties, byte[] body) throws IOException {
            String msg=new String(body,"utf-8");
            System.out.println("new!");
        }
    };
}

```

Recv [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86\_64\_1.6.0.u43\bin\javaw.exe (2018-1-26 下午11:35:34)
[recv] msg:hello simple !
[recv] msg:hello simple !
[recv] msg:hello simple !
[recv] msg:hello simple !

04.Java操作simple简单队列

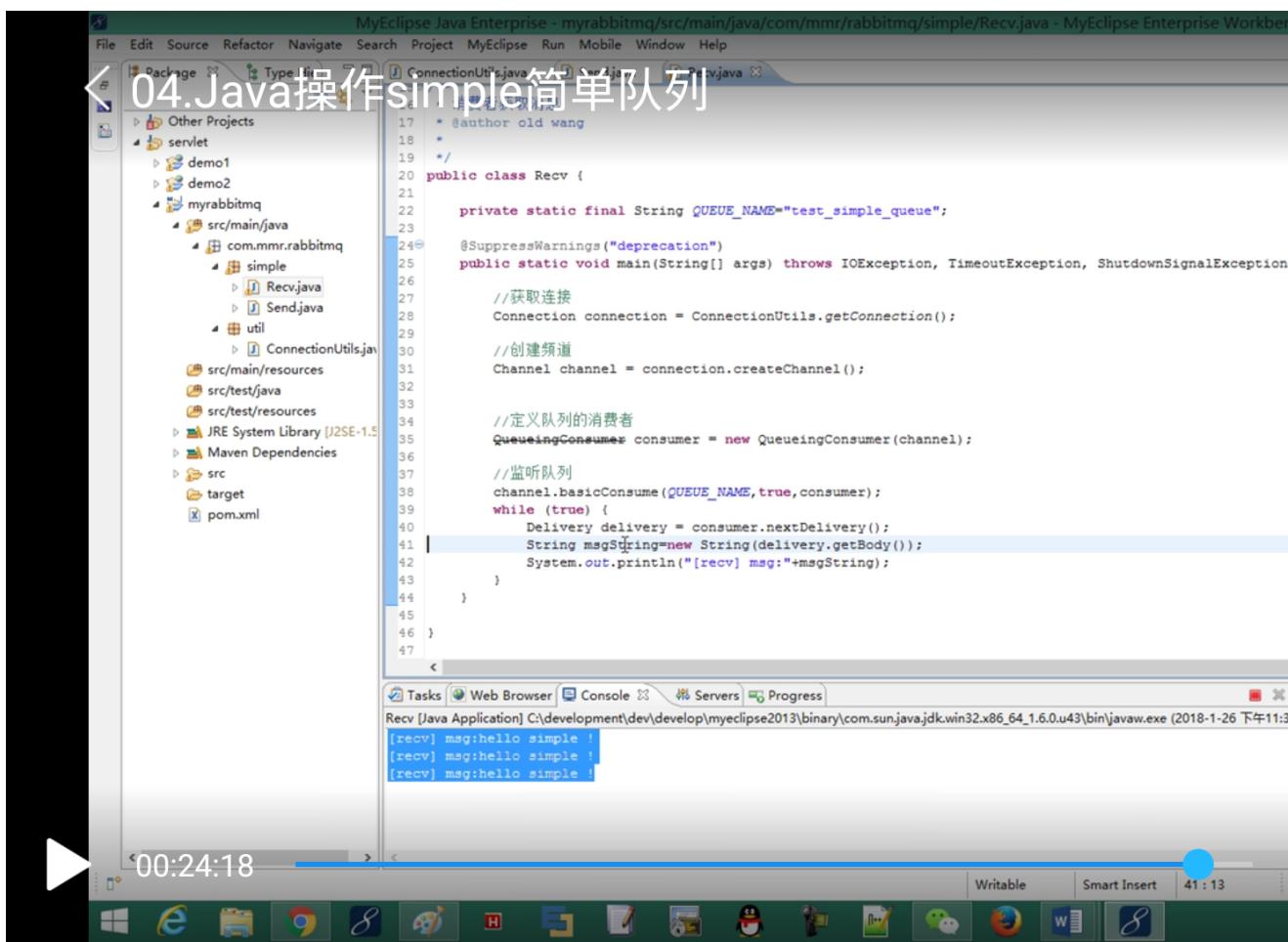
```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
```

```
17 * @author old wang
18 *
19 */
20 public class Recv {
21
22     private static final String QUEUE_NAME="test_simple_queue";
23
24     @SuppressWarnings("deprecation")
25     public static void main(String[] args) throws IOException, TimeoutException, ShutdownSignalException {
26
27         //获取连接
28         Connection connection = ConnectionUtils.getConnection();
29
30         //创建频道
31         Channel channel = connection.createChannel();
32
33         //定义队列的消费者
34         QueueingConsumer consumer = new QueueingConsumer(channel);
35
36         //监听队列
37         channel.basicConsume(QUEUE_NAME,true,consumer);
38
39         while (true) {
40             Delivery delivery = consumer.nextDelivery();
41             String msgString=new String(delivery.getBody());
42             System.out.println("[recv] msg:"+msgString);
43         }
44     }
45
46 }
47
```

```
Tasks Web Browser Console Servers Progress
```

```
Recv [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86_64_1.6.0.u43\bin\javaw.exe (2018-1-26 下午11:31)
[recv] msg:hello simple !
[recv] msg:hello simple !
[recv] msg:hello simple !
```

00:24:18



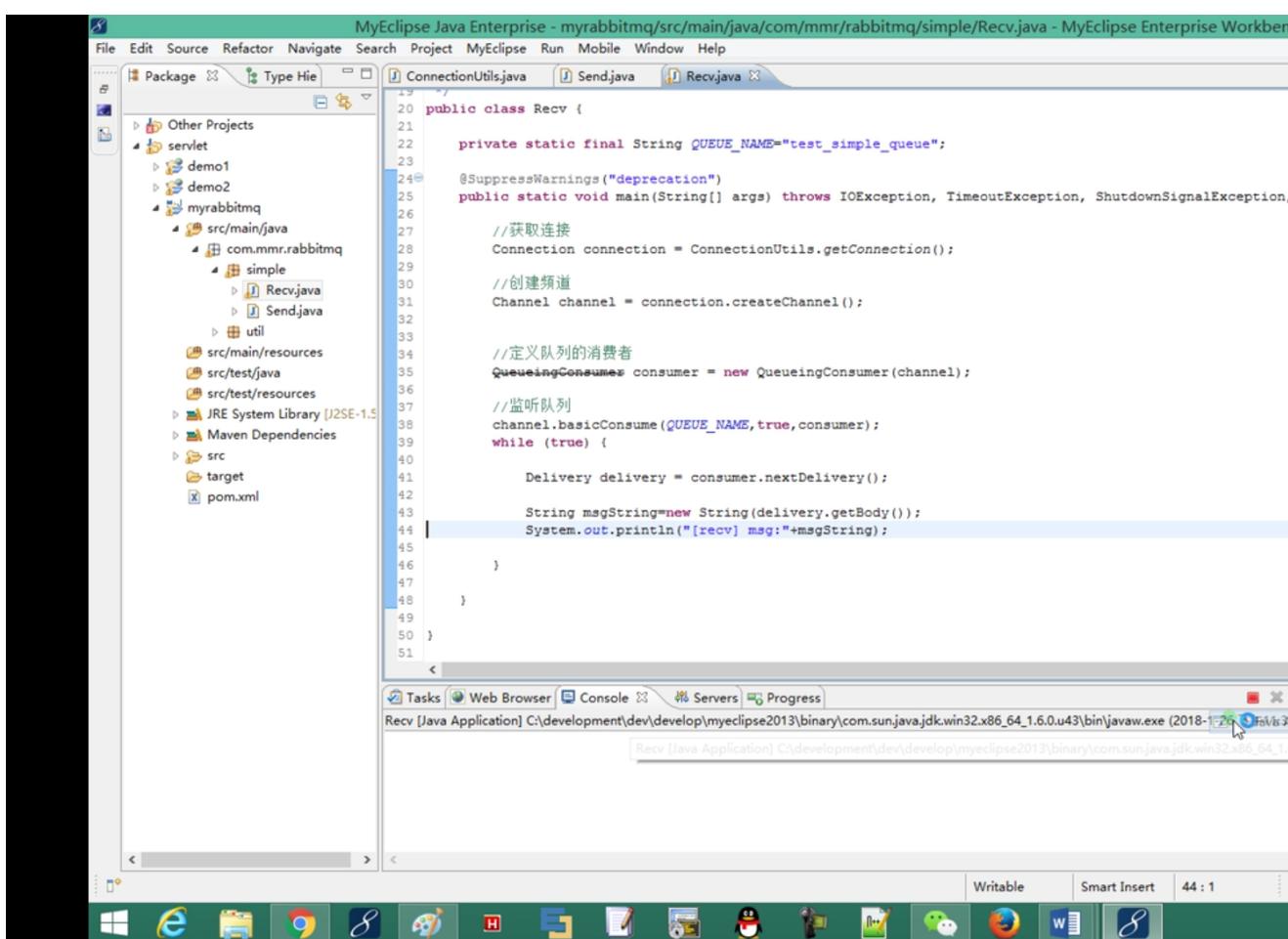
```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
```

```
19 */
20 public class Recv {
21
22     private static final String QUEUE_NAME="test_simple_queue";
23
24     @SuppressWarnings("deprecation")
25     public static void main(String[] args) throws IOException, TimeoutException, ShutdownSignalException {
26
27         //获取连接
28         Connection connection = ConnectionUtils.getConnection();
29
30         //创建频道
31         Channel channel = connection.createChannel();
32
33         //定义队列的消费者
34         QueueingConsumer consumer = new QueueingConsumer(channel);
35
36         //监听队列
37         channel.basicConsume(QUEUE_NAME,true,consumer);
38
39         while (true) {
40
41             Delivery delivery = consumer.nextDelivery();
42
43             String msgString=new String(delivery.getBody());
44             System.out.println("[recv] msg:"+msgString);
45
46         }
47
48     }
49
50 }
```

```
Tasks Web Browser Console Servers Progress
```

```
Recv [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86_64_1.6.0.u43\bin\javaw.exe (2018-1-26 下午11:31)
[recv] msg:hello simple !
[recv] msg:hello simple !
[recv] msg:hello simple !
```

44 : 1



MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/simple/Recv.java - MyEclipse Enterprise Workbench

## 04.Java操作simple简单队列

```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
```

Package Type Help

Other Projects

- servlet
  - demo1
  - demo2
- myrabbitmq
  - src/main/java
    - com.mmr.rabbitmq
      - simple
        - Recv.java
        - Send.java
      - util
    - src/main/resources
    - src/test/java
    - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
  - target
  - pom.xml

ConnectionUtils.java [Recv.java] 12:26 2018-01-05 10:18:53

```
1 import com.mmr.rabbitmq.util.ConnectionUtils;
2 import com.rabbitmq.client.Consumer;
3 import com.rabbitmq.client.TimeoutException;
4
5 /**
6  * 消费者获取消息
7  * @author old wang
8  */
9
10 public class Recv {
11
12     /**
13      * @param args
14      */
15
16     public static void main(String[] args) throws IOException, TimeoutException {
17         //获取连接
18         Connection connection = ConnectionUtils.getConnection();
19
20         //创建频道
21         Channel channel = connection.createChannel();
22
23         //声明队列
24         channel.queueDeclare("hello", false, false, false, null);
25
26         //消费
27         channel.basicConsume("hello", true, "hello", consumer);
28
29         System.out.println(" [*] Waiting for messages. To exit press CTRL+C");
30
31         //接收消息
32         consumer.receive();
33
34     }
35 }
```

Tasks Web Browser Console Servers Progress

<terminated> Send [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86\_64\_1.6.0.u43\bin\javaw.exe (2018-01-05 10:18:53)

--send msg:hello simple !

00:18:53

Writable Smart Insert 28 : 9

The screenshot shows the MyEclipse Java Enterprise IDE interface. The left pane displays the project structure under 'Other Projects'. The central pane shows the Java code for a 'Recv' class within the 'com.mmr.rabbitmq.simple' package. The code uses the RabbitMQ Java client library to declare a queue named 'hello' and consume messages from it. The right pane shows a terminal window with the command 'Send [Java Application] C:\development\dev\develop\myeclipse2013\binary\com.sun.java.jdk.win32.x86\_64\_1.6.0.u43\bin\javaw.exe (2018-01-05 10:18:53)' and the message '--send msg:hello simple !'. The status bar at the bottom indicates the time as '00:18:53'.

04.Java操作simple简单队列

```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
```

```
MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/simple/Send.java - MyEclipse Enterprise Work
```

```
1 ConnectionUtils.java
2
3 Other Projects
4 servlet
5   demo1
6   demo2
7 myrabbitmq
8   src/main/java
9     com.mmr.rabbitmq
10    simple
11      Send.java
12    util
13    src/main/resources
14    src/test/java
15    src/test/resources
16 JRE System Library [J2SE-1.5]
17 Maven Dependencies
18 src
19 target
20 pom.xml
```

```
4 import java.io.IOException;
5 import java.util.concurrent.TimeoutException;
6
7 import com.mmr.rabbitmq.util.ConnectionUtils;
8 import com.rabbitmq.client.Channel;
9 import com.rabbitmq.client.Connection;
10
11 public class Send {
12     private static final String QUEUE_NAME="test_simple_queue";
13     public static void main(String[] args) throws IOException, TimeoutException {
14         //获取一个连接
15         Connection connection = ConnectionUtils.getConnection();
16
17         //从连接中获取一个通道
18         Channel channel = connection.createChannel();
19         //创建队列声明
20         channel.queueDeclare(QUEUE_NAME, false, false, false, null);
21
22         String msg="hello simple !";
23
24         channel.basicPublish("", QUEUE_NAME, null, msg.getBytes());
25
26         channel.close();
27         connection.close();
28     }
29
30 }
31
32
33 }
```

```
Tasks Web Browser Console Servers Progress
```

```
No operations to display at this time.
```

```
00:15:46 Writable Smart Insert 31 : 1
```

04.Java操作simple简单队列

```
File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help
```

```
MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/simple/Send.java - MyEclipse Enterprise Work
```

```
1 ConnectionUtils.java
2
3 Other Projects
4 servlet
5   demo1
6   demo2
7 myrabbitmq
8   src/main/java
9     com.mmr.rabbitmq
10    simple
11      Send.java
12    util
13    src/main/resources
14    src/test/java
15    src/test/resources
16 JRE System Library [J2SE-1.5]
17 Maven Dependencies
18 src
19 target
20 pom.xml
```

```
2 package com.mmr.rabbitmq.simple;
3
4 import java.io.IOException;
5 import java.util.concurrent.TimeoutException;
6
7 import com.mmr.rabbitmq.util.ConnectionUtils;
8 import com.rabbitmq.client.Channel;
9 import com.rabbitmq.client.Connection;
10
11 public class Send {
12     private static final String QUEUE_NAME="test_simple_queue";
13     public static void main(String[] args) throws IOException, TimeoutException {
14         //获取一个连接
15         Connection connection = ConnectionUtils.getConnection();
16
17         //从连接中获取一个通道
18         Channel channel = connection.createChannel();
19         //创建队列声明
20         channel.queueDeclare(QUEUE_NAME, false, false, false, null);
21
22         String msg="hello simple !";
23
24         channel[1];
25
26     }
27
28
29
30
31
32 }
```

```
Tasks Web Browser Console Servers Progress
```

```
No operations to display at this time.
```

```
00:14:48 Writable Smart Insert 26 : 16
```

04.Java操作simple简单队列

```
MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/util/ConnectionUtils.java - MyEclipse Enterprise Workbench
```

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help

Package pom.xml ConnectionUtils.java

```
9 public class ConnectionUtils {  
10     /**  
11      * 获得MQ的连接  
12      * @return  
13      * @throws TimeoutException  
14      * @throws IOException  
15      */  
16     public static Connection getConnection() throws IOException, TimeoutException{  
17         //定义一个连接工厂  
18         ConnectionFactory factory =new ConnectionFactory();  
19         //设置服务地址  
20         factory.setHost("127.0.0.1");  
21         //AMQP 5672  
22         factory.setPort(5672);  
23         //vhost  
24         factory.setVirtualHost("/vhost_mmr");  
25         //用户名  
26         factory.setUsername("user_mmr");  
27         //密码  
28         factory.setPassword("123");  
29         return factory.newConnection();  
30     }  
31 }  
32  
33  
34  
35  
36  
37  
38  
39
```

Tasks Web Browser Console Servers Progress

No operations to display at this time.

00:09:21

Writable Smart Insert 10:1

This screenshot shows the MyEclipse Java Enterprise IDE interface. The title bar indicates the project is 'myrabbitmq' and the file is 'src/main/java/com/mmr/rabbitmq/util/ConnectionUtils.java'. The code editor displays Java code for creating a RabbitMQ connection. The code defines a static method 'getConnection()' that creates a 'ConnectionFactory' object, sets its host to '127.0.0.1', port to 5672, and virtual host to '/vhost\_mmr'. It then returns a new connection. The code uses standard Java annotations like 'throws' for exceptions. Below the code editor is a toolbar with various icons for tasks like web browser, console, servers, and progress. The bottom status bar shows the current time as '00:09:21' and status indicators for 'Writable' and 'Smart Insert'.

```
MyEclipse Java Enterprise - myrabbitmq/src/main/java/com/mmr/rabbitmq/util/ConnectionUtils.java - MyEclipse Enterprise Workbench
```

File Edit Source Refactor Navigate Search Project MyEclipse Run Mobile Window Help

Package pom.xml ConnectionUtils.java

```
1 package com.mmr.rabbitmq.util;  
2  
3 import com.rabbitmq.client.Connection;  
4 import com.rabbitmq.client.ConnectionFactory;  
5  
6 public class ConnectionUtils {  
7  
8     /**  
9      * 获得MQ的连接  
10     * @return  
11     */  
12     public static Connection getConnection(){  
13         //定义一个连接工厂  
14         ConnectionFactory factory =new ConnectionFactory();  
15         //设置服务地址  
16         factory.setHost("127.0.0.1");  
17         //AMQP 5672  
18         factory.setPort(5672);  
19         //vhost  
20         factory.setVirtualHost("/vhost_mmr");  
21         //用户名  
22         factory.setUsername("user_mmr");  
23         //密码  
24         factory.setPassword("123");  
25         //返回  
26         return null;  
27     }  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39
```

Tasks Web Browser Console Servers Progress

No operations to display at this time.

27:38

This screenshot shows the same MyEclipse Java Enterprise IDE interface as the previous one, but with a different state. The code editor now shows a modified version of the 'getConnection()' method. The last line of code, 'return null;', has been added. The rest of the code remains the same. The bottom status bar shows the current time as '27:38'.

MyEclipse Java Enterprise - myrabbitmq/pom.xml - MyEclipse Enterprise Workbench

```
<dependency>
    <groupId>com.rabbitmq</groupId>
    <artifactId>amqp-client</artifactId>
    <version>4.0.2</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.7.10</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>1.7.5</version>
</dependency>
<dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
</dependency>
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
</dependency>
</dependencies>
```

