<https://www.cnblogs.com/xu-lei/p/MQ.html>

[**java实现MQ消息收发两种方式**](http://www.cnblogs.com/xu-lei/p/MQ.html)

定义：

消息队列（MQ）是一种应用程序对应用程序的通信方法。应用程序通过写和检索出入列队的针对应用程序的数据（消息）来通信，而无需专用连接来链接它们。简单理解：蓝牙配对

jar包依赖：

<!-- 加入mq消息依赖包 -->  
  <dependency>  
         <groupId>com.rabbitmq</groupId>  
         <artifactId>amqp-client</artifactId>  
         <version>3.0.4</version>  
  </dependency>

方式一：

（发）

[复制代码](javascript:void(0);)

import java.io.IOException;

import com.rabbitmq.client.Channel;

import com.rabbitmq.client.Connection;

import com.rabbitmq.client.ConnectionFactory;

public class Send {

private final static String QUEUE\_NAME = "qq";

public static void main(String[] args) throws IOException {

ConnectionFactory factory = new ConnectionFactory();

factory.setHost("hostname");

factory.setUsername("username");

factory.setPassword("password");

factory.setPort(5672);

Connection connection = factory.newConnection();

Channel channel = connection.createChannel();

channel.queueDeclare(QUEUE\_NAME, false, false, false, null);

String message = "Hello World!";

channel.basicPublish("", QUEUE\_NAME, null, message.getBytes());

System.out.println(" [x] Sent '" + message + "'");

channel.close();

connection.close();

}

}

[复制代码](javascript:void(0);)

(收)

[复制代码](javascript:void(0);)

import com.rabbitmq.client.ConnectionFactory;

import com.rabbitmq.client.QueueingConsumer;

import com.rabbitmq.client.Channel;

import com.rabbitmq.client.Connection;

public class Reqv {

private final static String QUEUE\_NAME = "qq";

public static void main(String[] argv) throws Exception {

ConnectionFactory factory = new ConnectionFactory();

factory.setUsername("username");

factory.setPassword("password");

factory.setHost("hostname");

factory.setVirtualHost("/");

factory.setPort(5672);

Connection connection = factory.newConnection();

Channel channel = connection.createChannel();

channel.queueDeclare(QUEUE\_NAME, false, false, false, null);

System.out.println(" [\*] Waiting for messages. To exit press CTRL+C");

QueueingConsumer consumer = new QueueingConsumer(channel);

channel.basicConsume(QUEUE\_NAME, true, consumer);

while(true){

QueueingConsumer.Delivery delivery = consumer.nextDelivery();

String message = new String(delivery.getBody());

System.out.println(" [x] Received '" + message + "'");

}

}

}

[复制代码](javascript:void(0);)

方式二：

（发）

[复制代码](javascript:void(0);)

import com.rabbitmq.client.Channel;

import com.rabbitmq.client.Connection;

import com.rabbitmq.client.ConnectionFactory;

public class Send2 {

private static final String EXCHANGE\_NAME = "qq" ;

public static void main (String [] argv) throws java.io.IOException {

ConnectionFactory factory = new ConnectionFactory();

factory.setHost("hostname");

factory.setUsername("username");

factory.setPassword("password");

factory.setPort(5672);

Connection connection = factory.newConnection();

Channel channel = connection.createChannel();

channel.exchangeDeclare(EXCHANGE\_NAME,"fanout",true);

String message = "hello world";

channel.basicPublish(EXCHANGE\_NAME,"",null,message.getBytes());

System.out.println("[x] Sent'" + message + "'");

channel.close();

connection.close();

}

[复制代码](javascript:void(0);)

（收）

[复制代码](javascript:void(0);)

import java.io.IOException;import com.alibaba.fastjson.JSONObject;

import com.rabbitmq.client.AMQP;

import com.rabbitmq.client.Channel;

import com.rabbitmq.client.Connection;

import com.rabbitmq.client.ConnectionFactory;

import com.rabbitmq.client.Consumer;

import com.rabbitmq.client.DefaultConsumer;

import com.rabbitmq.client.Envelope;

public class Reqv2 {

private static final String EXCHANGE\_NAME = "qq" ;public static void main (String [] argv) throws Exception {

ConnectionFactory factory = new ConnectionFactory();

factory.setHost("hostname");

factory.setUsername("username");

factory.setPassword("password");

factory.setPort(5672);

Connection connection = factory.newConnection();

Channel channel = connection.createChannel();

channel.exchangeDeclare(EXCHANGE\_NAME,"fanout",true);

String queueName = channel.queueDeclare().getQueue();

channel.queueBind(queueName,EXCHANGE\_NAME,"");

System.out.println("[\*]等待消息。退出按CTRL + C");

Consumer consumer = new DefaultConsumer(channel){

@Override

public void handleDelivery (String consumerTag,Envelope envelope,AMQP.BasicProperties properties,byte[] body) throws IOException {

String message = new String(body,"UTF-8");

System.out.println("[x] Received'" + message + "'");

}

};

channel.basicConsume(queueName,true,consumer);

}

}

[复制代码](javascript:void(0);)

参考网址：

http://www.rabbitmq.com/tutorials/tutorial-one-java.html