

（四）基于myeclipse的RocketMQ--Demo实践

接上文，搭建好环境，用example中的示例只能进行有限的测试任务。Rocket-console无法模拟发送和接收消息，所以自定义测试任务需要自行编写demo程序。

作者：Nick

创建Demo项目流程

1. 下载myeclipse

2. 安装maven环境，关联到myeclipse

myeclipse 添加自定义jdk环境： [参考文章A](#)

myeclipse添加自定义maven环境： [参考文章B](#) 、 [参考文章C](#)

3. 创建maven项目，配置pom.xml

File--New--Other--Maven Project--(Create a simple project)

4. 导入依赖包

直接把 `RocketMQ/devenv/lib` 下的jar包都copy到刚创建的maven项目内

5. 配置pom.xml

直接把RocketMQ的pom.xml的内容copy过去

6. 编写消息生产者Producer

src--New Package--New Class--Producer.java

内容可以参考 `com.alibaba.rocketmq.example.quickstart(simple)` 下的Producer，下文的Consumer类似

```

package com.alibaba.rocketmq.example.quickstart;

import com.alibaba.rocketmq.client.exception.MQClientException;
import com.alibaba.rocketmq.client.producer.DefaultMQProducer;
import com.alibaba.rocketmq.client.producer.LocalTransactionExecuter;
import com.alibaba.rocketmq.client.producer.LocalTransactionState;
import com.alibaba.rocketmq.client.producer.SendResult;
import com.alibaba.rocketmq.common.message.Message;
import com.alibaba.rocketmq.remoting.common.RemotingHelper;

public class Producer {
    public static void main(String[] args) throws MQClientException, InterruptedException {
        // tc_pro1为Producer group name
        DefaultMQProducer producer = new DefaultMQProducer("tc_pro1");
        // 手动指定Namesrv服务地址
        producer.setNamesrvAddr("192.168.1.170:9876");
        producer.setInstanceName("Producer1-tp1");
        producer.start();

        // 如果broker关闭了自动创建Topic功能，请手动添加Topic: tc_demo，以确保能正常发送消息
        for (int i = 0; i < 1; i++) {
            try {
                Message msg = new Message("tc_demo", // topic
                    "TagA", // tag
                    ("Hello RocketMQ " + i).getBytes(RemotingHelper.DEFAULT_CHARSET) // body
                );
                SendResult sendResult = producer.send(msg);
                LocalTransactionExecuter tranExecuter = new LocalTransactionExecuter() {

                    public LocalTransactionState executeLocalTransactionBranch(Message msg,
Object arg) {
                        // TODO Auto-generated method stub
                        return null;
                    }
                };

                //producer.sendMessageInTransaction(msg, tranExecuter, arg)
                System.out.println(sendResult);
            } catch (Exception e) {
                e.printStackTrace();
                Thread.sleep(1000);
            }
        }

        producer.shutdown();
    }
}

```

7.编写消息消费者Consumer

xxx Package--New Class--Consumer.java

```

package com.alibaba.rocketmq.example.quickstart;

import com.alibaba.rocketmq.client.consumer.DefaultMQPushConsumer;
import com.alibaba.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
import com.alibaba.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
import com.alibaba.rocketmq.client.consumer.listener.MessageListenerConcurrently;
import com.alibaba.rocketmq.client.exception.MQClientException;
import com.alibaba.rocketmq.common.consumer.ConsumeFromWhere;
import com.alibaba.rocketmq.common.message.MessageExt;

import java.util.List;

public class Consumer1 {

    public static void main(String[] args) throws InterruptedException, MQClientException {
        // tc_con1为Consumer group name,如果broker关闭了自动订阅功能,请手动添加订阅tc_con1,以确保能
        正常接收消息
        DefaultMQPushConsumer consumer = new DefaultMQPushConsumer("tc_con1");

        consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_FIRST_OFFSET);
        // 手动指定Namesrv服务地址
        consumer.setNamesrvAddr("192.168.1.170:9876");
        consumer.setInstanceName("Consumer1-tp1");

        consumer.subscribe("tc_demo", "*");

        consumer.registerMessageListener(new MessageListenerConcurrently() {

            public ConsumeConcurrentlyStatus consumeMessage(List msgs, ConsumeConcurrentlyContext
context) {
                System.out.println(Thread.currentThread().getName() + " Receive New Messages: " +
msgs);
                return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
            }
        });

        consumer.start();

        System.out.println("Consumer Started.");
    }
}

```

8.启动Consumer，Producer进行消息收发

前提：环境搭建成功，Namesrv 和 Broker服务运行正常，可通过 `jps` 查看服务是否运行

run Consumer.java /Producer.java 从myeclipse--console可以看到Consumer角色成功启动、Producer消息发送、Consumer接收消息。

至此，基于myeclipse上RocketMQ的demo实践流程就走通了，更多的自定义扩展可以参考其项目源码

参考文章D: [Producer多topic发送，Consumer多topic消费](#)

FAQ

1.win10下安装maven完成后，`mvn -version`显示报错

```
Error: JAVA_HOME is set to an invalid directory.JAVA_HOME = "C:\Program Files\Java\jdk1.7.0_17\bin"Please set the JAVA_HOME variable in your environment to match the location of your Java installation.
```

解决方案：jdk,maven的环境变量虽已在path里设置完成，且jdk正常。但maven启动另需JAVA_HOME，所以手动添加JAVA_HOME的值：xxx/java/jdk_1.7.xx（no /bin）

2.Producer发送信息失败或Consumer无法接受信息

问题起因和解决方案：

1. Namesrv地址未指定或错误，请确认Namesrv地址
2. Namesrv或Broker未启动，通过 `jps` 查询集群（单机）节点服务状态，如果没有NamesrvStartup和BrokerStartup，重新启动（可以参看系列文章（二）/（三））
3. Broker关闭的自动创建topic和自动订阅消费组的功能。调用mqadmin 下的 updateTopic 或 updateSubGroup 创建topic或订阅组