rocketmq-client-dotnet

.NET客户端API

重构说明

代码逻辑说明

image

项目地址

ClementIV/rocketmq-client-dotnet

项目目录简介

```
.
├── rocketmq-client-dotnet RocketMQ .NET Client 文件
    └── example
        └── quickStart
            ├── ConsumerQuickStart PushConsumer Quick Start
            ├── ProducerQuickStart Producer Quick Start
            └── PullConsumerQuickStart PullConsumer Quick Start
└── src
     └── RocketMQ.NETClient  NET Client
        ├── Consumer Consumer API
        ├── Interop  常量等
        ├── Message Message API
        └── Producer Producer API
```

调试环境快速部署

TODO

API 对齐说明

功能	C	.NET
同步消息发送	Y	Y
顺序消息发送	Y	Y
单向消息发送	Y	Y
拉取消息消费	Y	Y
推送消息消费	Y	Y

功能	C	.NET
延时消息	Y	Y
消息压缩	Y	Y
消息过滤	Y	Y
字符串消息体	Y	Y
字节流消息体	N	N
Topic设置	Y	Y

Producer 创建说明

之前的方式

1. 创建`DefaultProducerBuilder`对象 `producerBuilder`
2. 使用 `producerBuilder`设置想要生成的`producerBuilder`参数
3. 调用 `producerBuilder`中的`Builder`函数返回一个`IProducer` 实例 `producer`
4. 使用 `producer`

示例代码：

```
    //创建一个 producerBuilder
    DefaultProducerBuilder producerBuilder = new  DefaultProducerBuilder("group name",null,null);

    //设置想要生成的 producerBuilder 参数
    producerBuilder = producerBuilder.SetProducerNameServerAddress("127.0.0.1:9876");
    //··· 其他的一些设置

    // 函数返回一个 IProducer 实例 producer
    IProducer producer = producerBuilder.Builder();

    // 使用producer发送消息
    producer.StartProducer();

    var sendResult =   producer.SendMessageSync(producer, messageIntPtr, out CSendResult sendResultStruct);
                    Console.WriteLine("send result:" + sendResult + ", msgId: " + sendResultStruct.msgId.ToString());
```

重构使用说明

1. 创建producer对象(多种构造函数)
2. 使用producer发送消息

```
    //创建一个 producer
    MQProducer producer = new MQProducer("GroupA", "127.0.0.1:9876");
    producer.StartProducer();

    // 创建一个消息 message
    MQMessage message = new MQMessage("test");

    // 使用producer发送消息
    // SendMessageSync
    var sendResult = producer.SendMessageSync(message);
    Console.WriteLine("send result:" + sendResult + ", msgId: " +
sendResult.MessageId);
```

对应Producer项目

ProducerQuickStart

PushConsumer使用说明(推荐使用)

使用说明

1. 创建一个Push消费者
2. 订阅一个`topic`
3. 注册回调函数
4. 启动消费者

```
   // 创建一个Push消费者
    var consumer = new MQPushConsumer("xx", "127.0.0.1:9876");
    Console.WriteLine($"consumer: {consumer}");

    // 设置日志目录和级别
    consumer.SetPushConsumerLogPath(".\\consumer_log.txt");
    consumer.SetPushConsumerLogLevel(LogLevel.Trace);

    // 获取消费者组号
    var groupId = consumer.GetPushConsumerGroupID();
    Console.WriteLine($"groupId: {groupId}");

    //  订阅一个`topic`
    consumer.Subscribe("test", "*");

    //注册回调函数
    consumer.RegisterMessageCallback(_callback);

    //启动消费者
    var result=consumer.StartPushConsumer();
    Console.WriteLine($"start push consumer ptr: {result}");
```

对应项目

ConsumerQuickStart

PullConsumer 使用说明

使用说明

1. 创建一个PullConsumer
2. 开启消费者
3. 填充消息队列
4. 主动拉取消费

```
    //创建一个PullConsumer
    MQPullConsumer consumer = new MQPullConsumer("xxx", "127.0.0.1:9876", ".\\log.txt", LogLevel.Trace);

    //开启消费者
    var result = consumer.StartPullConsumer();
    Console.WriteLine($"start Pull consumer ? : {result}");

    //填充消息队列
    CMessageQueue[] msgs = consumer.FetchSubscriptionMessageQueues("test");


    for (int j = 0; j < msgs.Length; j++)
    {
        int flag = 0;


        Console.WriteLine("msg topic : " + new string(msgs[j].topic));

        MessageQueue mq = new MessageQueue { topic = new string(msgs[j].topic), brokeName = new string(msgs[j].brokerName), queueId = msgs[j].queueId };

        while (true)
        {
            try
            {

                //主动拉取消费
                CPullResult cPullResult = consumer.Pull(mq,msgs[j], "", MQPullConsumer.getMessageQueueOffset(mq), 32);
                Console.WriteLine(new string(msgs[j].topic) + " status : " + cPullResult.pullStatus +"Max offset "+ cPullResult.maxOffset + " offset: " + cPullResult.nextBeginOffset + " Quene Id" + msgs[j].queueId);
                //Console.WriteLine(" " + msg.topic);
                long a = cPullResult.nextBeginOffset;
                MQPullConsumer.putMessageQueueOffset(mq, a);
                switch (cPullResult.pullStatus)
                {
                    case CPullStatus.E_FOUND:
                        break;
                    case CPullStatus.E_NO_MATCHED_MSG:
                                break;
                    case CPullStatus.E_NO_NEW_MSG:
                        flag = 1;
                        break;
                    case CPullStatus.E_OFFSET_ILLEGAL:
                        flag = 2;
                        break;
                    default:
                        break;
                }

                if(flag == 1|| cPullResult.nextBeginOffset ==
cPullResult.maxOffset)
                {
                    break;
                }
                if (flag == 2)
                {
                    break;
                }

            }
            catch (Exception e)
            {
                Console.WriteLine(e.Message);
            }
        }


    }
```

项目地址

PullConsumerQuickStart