



链滴

## RocketMq (二) 简单的消息示例

作者: [sulinfyx](#)

原文链接: <https://ld246.com/article/1512875788488>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

使用RocketMQ以三种方式发送消息：可靠的同步，可靠的异步和单向传输。

这个页面举例说明了这三种消息发送方式。 检查笔记和示例，找出哪种方式用于您的特定用例。

## 可靠的同步传输

应用：可靠的同步传输广泛应用于重要通知消息，短信通知，短信营销系统等。

```
public class SyncProducer {
    public static void main(String[] args) throws Exception {
        //Instantiate with a producer group name.
        DefaultMQProducer producer = new
            DefaultMQProducer("please_rename_unique_group_name");
        //Launch the instance.
        producer.start();
        for (int i = 0; i < 100; i++) {
            //Create a message instance, specifying topic, tag and message body.
            Message msg = new Message("TopicTest" /* Topic */,
                "TagA" /* Tag */,
                ("Hello RocketMQ " +
                    i).getBytes(RemotingHelper.DEFAULT_CHARSET) /* Message body */
            );
            //Call send message to deliver message to one of brokers.
            SendResult sendResult = producer.send(msg);
            System.out.printf("%s\n", sendResult);
        }
        //Shut down once the producer instance is not longer in use.
        producer.shutdown();
    }
}
```

## 可靠的异步传输

应用：异步传输一般用于响应时间敏感的业务场景。

```
public class AsyncProducer {
    public static void main(String[] args) throws Exception {
        //Instantiate with a producer group name.
        DefaultMQProducer producer = new DefaultMQProducer("ExampleProducerGroup");
        //Launch the instance.
        producer.start();
        producer.setRetryTimesWhenSendAsyncFailed(0);
        for (int i = 0; i < 100; i++) {
            final int index = i;
            //Create a message instance, specifying topic, tag and message body.
            Message msg = new Message("TopicTest",
                "TagA",
                "OrderID188",
                "Hello world".getBytes(RemotingHelper.DEFAULT_CHARSET));
            producer.send(msg, new SendCallback() {
                @Override
                public void onSuccess(SendResult sendResult) {
                    System.out.printf("%-10d OK %s %n", index,
```

```

        sendResult.getMsgId());
    }
    @Override
    public void onException(Throwable e) {
        System.out.printf("%-10d Exception %s %n", index, e);
        e.printStackTrace();
    }
    });
}
//Shut down once the producer instance is not longer in use.
producer.shutdown();
}
}

```

## 单向传输

应用：单向传输用于需要中等可靠性的情况，例如日志收集。

```

public class OnewayProducer {
    public static void main(String[] args) throws Exception{
        //Instantiate with a producer group name.
        DefaultMQProducer producer = new DefaultMQProducer("ExampleProducerGroup");
        //Launch the instance.
        producer.start();
        for (int i = 0; i < 100; i++) {
            //Create a message instance, specifying topic, tag and message body.
            Message msg = new Message("TopicTest" /* Topic */,
                "TagA" /* Tag */,
                ("Hello RocketMQ " +
                    i).getBytes(RemotingHelper.DEFAULT_CHARSET) /* Message body */
            );
            //Call send message to deliver message to one of brokers.
            producer.sendOneway(msg);
        }
        //Shut down once the producer instance is not longer in use.
        producer.shutdown();
    }
}

```