<http://kafka.apache.org/quickstart>

官网大致介绍了一下，安装使用，集群搭建（在一个机器上）。

使用kafka connect 来导入和导出数据

bin/connect-standalone.sh config/connect-standalone.properties config/connect-file-source.properties config/connect-file-sink.properties

config/connect-file-source.properties 源数据文件的配置

config/connect-file-sink.properties 输出数据的文件配置

 Once the Kafka Connect process has started, the source connector should start reading lines from test.txt and producing them to the topic connect-test, and the sink connector should start reading messages from the topic connect-test and write them to the file test.sink.txt

Kafka stream处理数据

一个统计消息单词数量的功能。

从一个topic（streams-plaintext-input）中获取数据，

通过org.apache.kafka.streams.examples.wordcount.WordCountDemo计算，

然后往另一个topic（streams-wordcount-output）写数据。

其中比较关键的是kTable，KStream

这个程序是不间断的。

bin/kafka-run-class.sh org.apache.kafka.streams.examples.wordcount.WordCountDemo

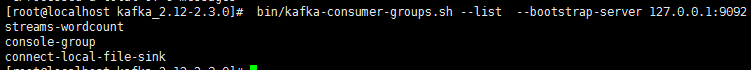
consumer的group不用单独创建，好像没有创建的命令或者api

bin/kafka-console-consumer.sh --bootstrap-server localhost:9093,localhost:9094 --topic test --from-beginning --group console-group

--group console-group 表示使用 console-group

可以可看console-group有了木有

bin/kafka-consumer-groups.sh --list --bootstrap-server 127.0.0.1:9092



手动提交offset。比较灵活，可以在处理成功消息后，再提交。这样避免未成功处理，也提交offset而丢失掉消息。但是也有重复消费消息的危险，比如处理成功后还未提交offset。这时可以用at-last-once。each record will likely be delivered one time but in failure cases could be duplicated

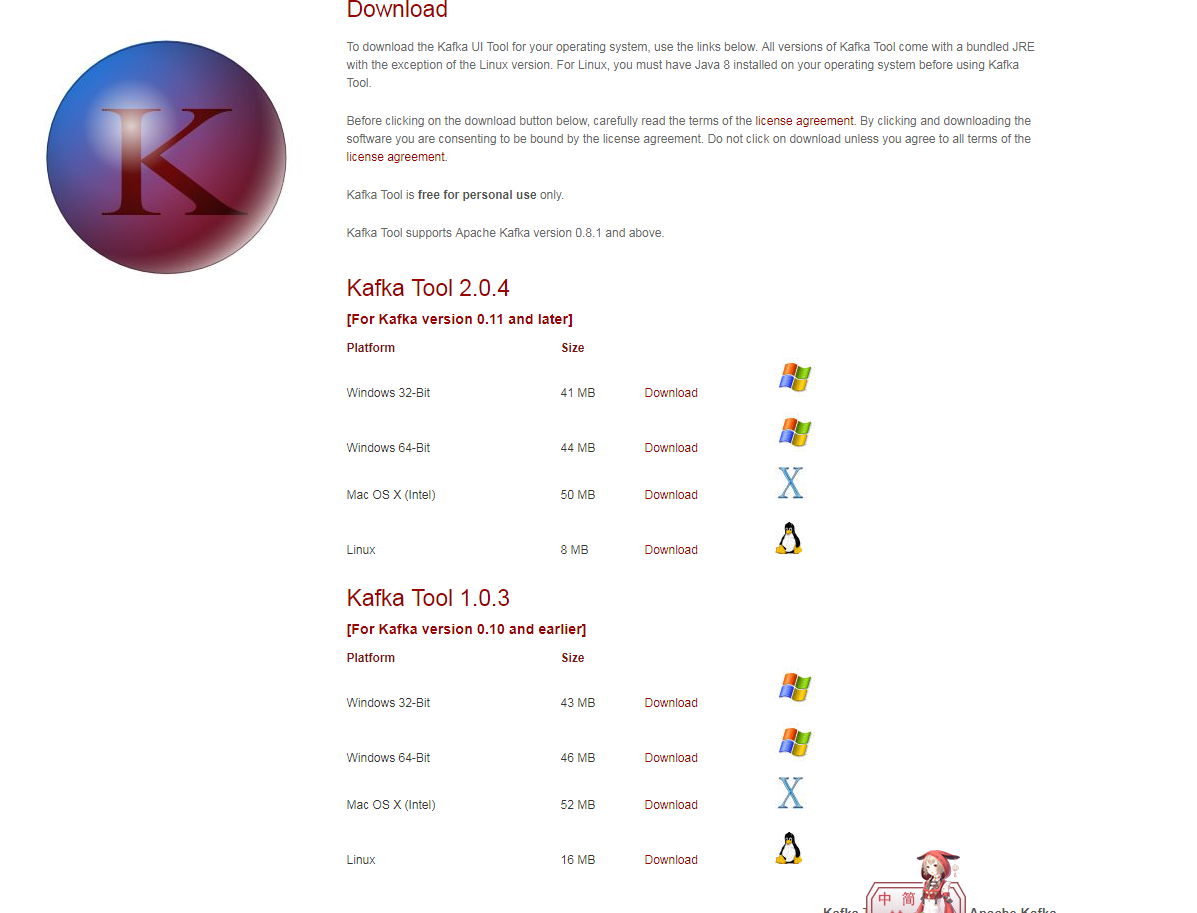
Using automatic offset commits can also give you "at-least-once" delivery, but the requirement is that you must consume all data returned from each call to [poll(Duration)](http://kafka.apache.org/23/javadoc/org/apache/kafka/clients/consumer/KafkaConsumer.html#poll-java.time.Duration-) before any subsequent calls, or before [closing](http://kafka.apache.org/23/javadoc/org/apache/kafka/clients/consumer/KafkaConsumer.html#close--) the consumer. If you fail to do either of these, it is possible for the committed offset to get ahead of the consumed position, which results in missing records. The advantage of using manual offset control is that you have direct control over when a record is considered "consumed."

Kafka java-consumer api的地址，介绍java-consumer怎么对接kafka

<http://kafka.apache.org/23/javadoc/index.html?org/apache/kafka/clients/consumer/KafkaConsumer.html>

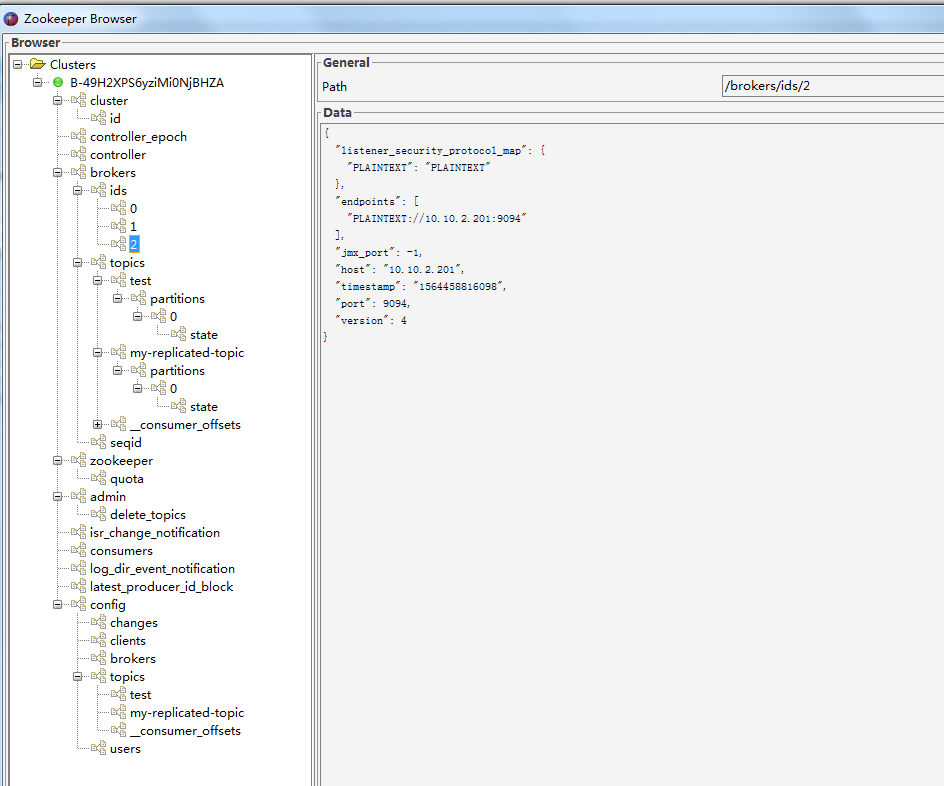
kafka的可视化工具<http://www.kafkatool.com/download.html>

kafka tool



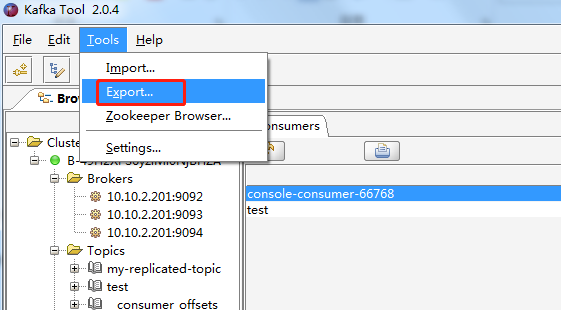
用法可以参考官网说明 <http://www.kafkatool.com/features.html>

可以查看zookeeper中的数据，看起来比较清楚

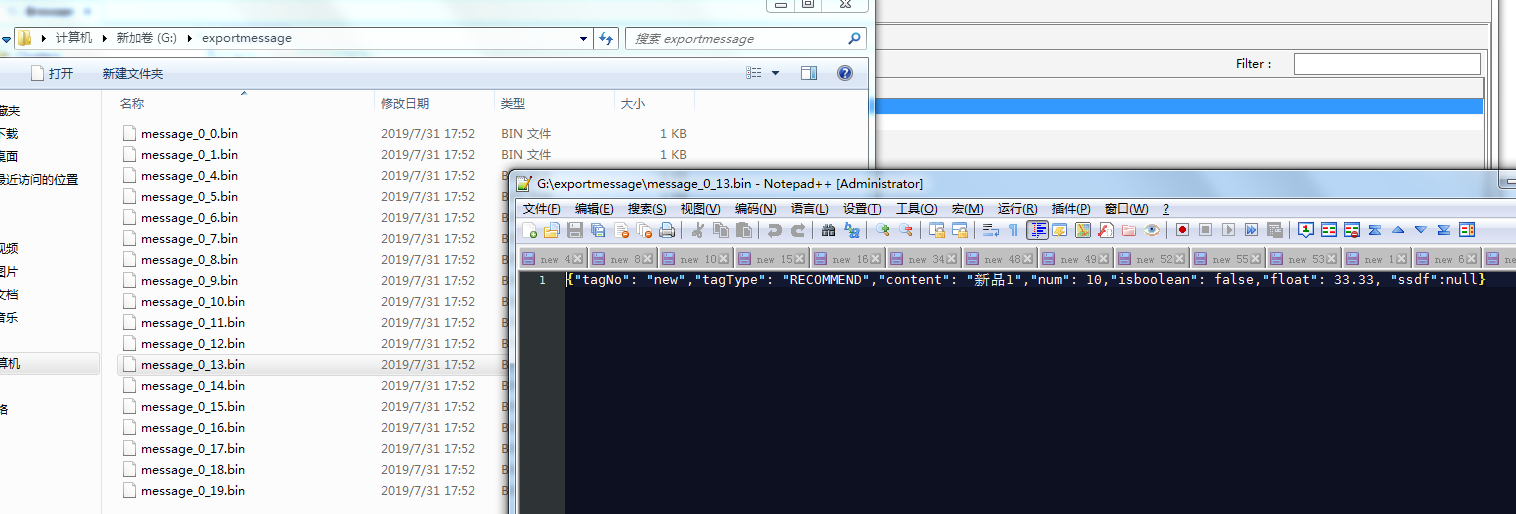


单纯的zookeeper可视化工具zkui，在github上面有这么一个工程。

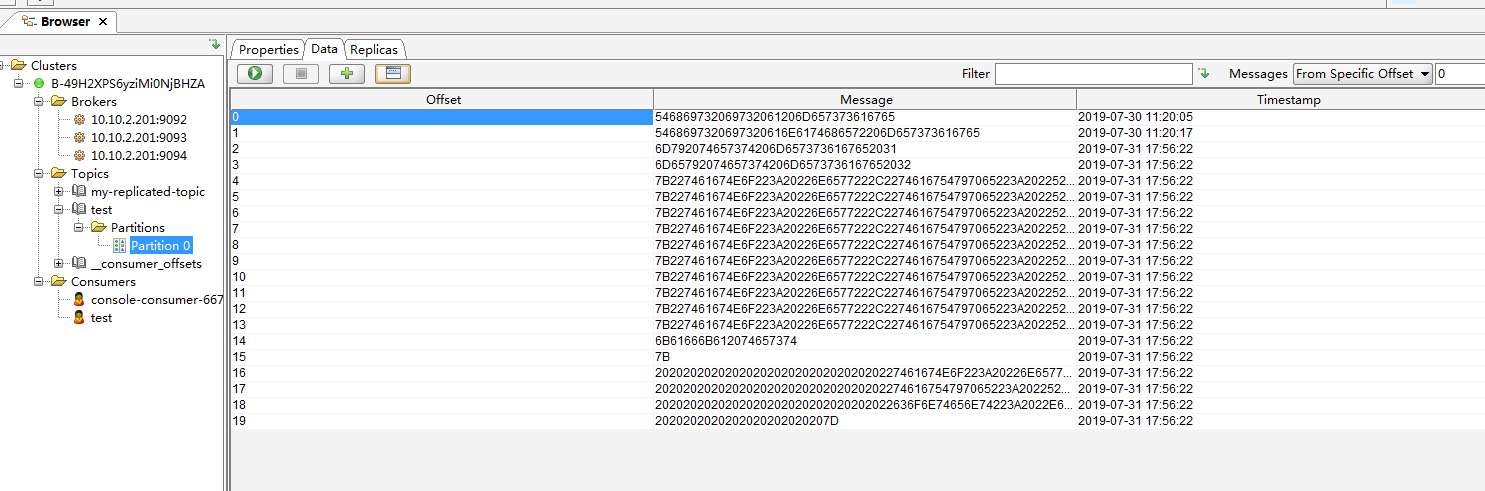
可以导出消息



导出的消息，一条消息一个文件

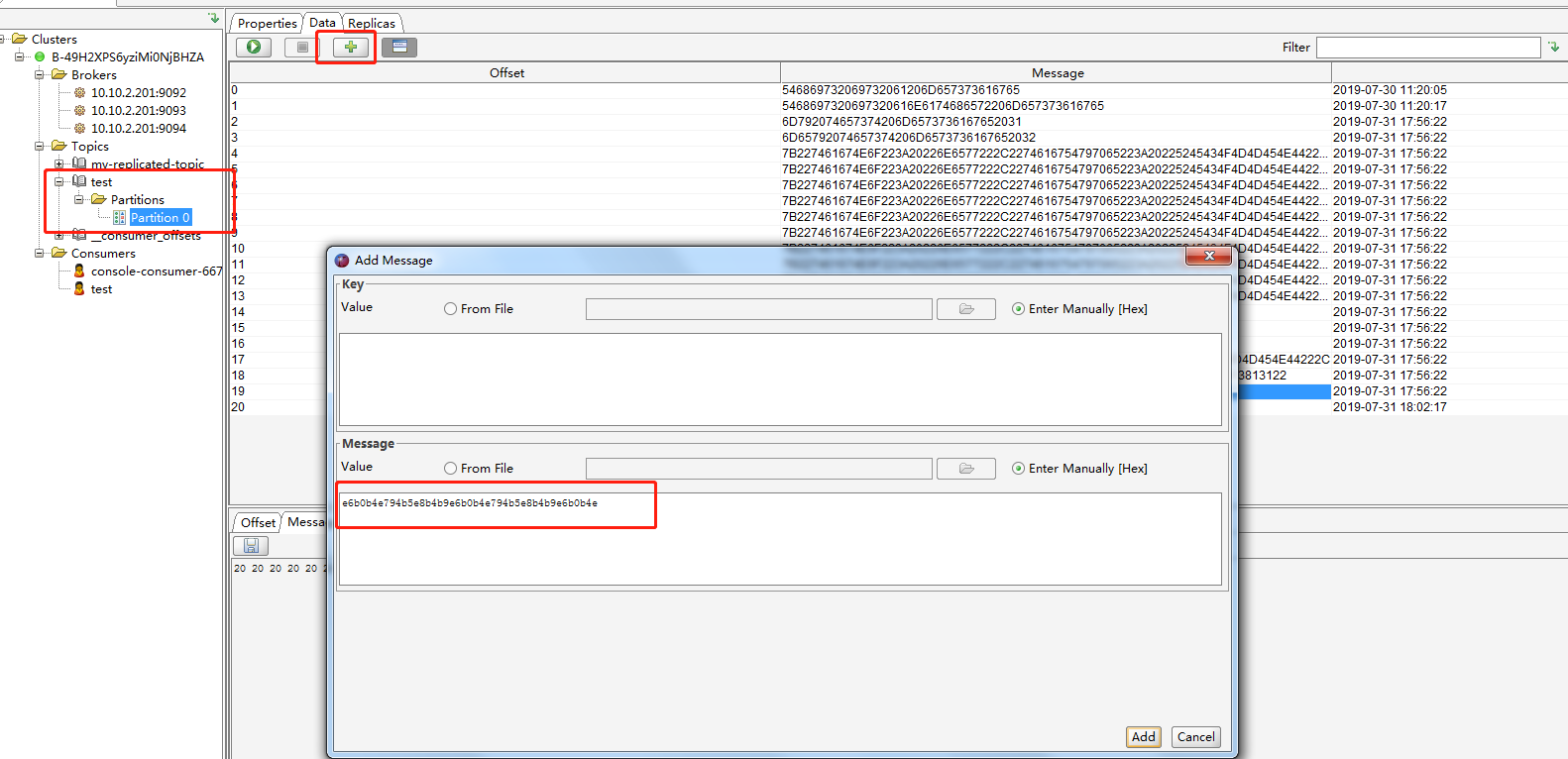


可以查看kafka集群中topic下面的消息



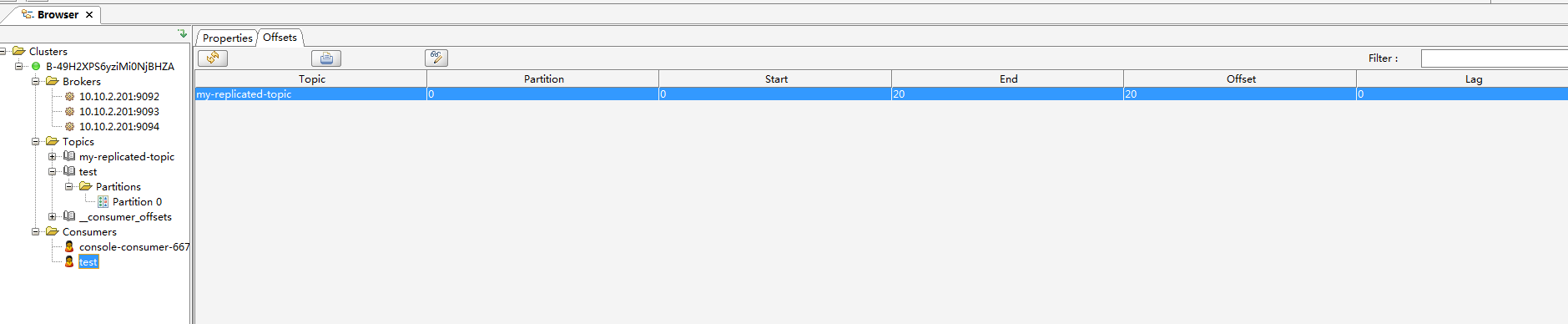
其中的消息message字段是16进制编码过的消息内容

可以添加消息

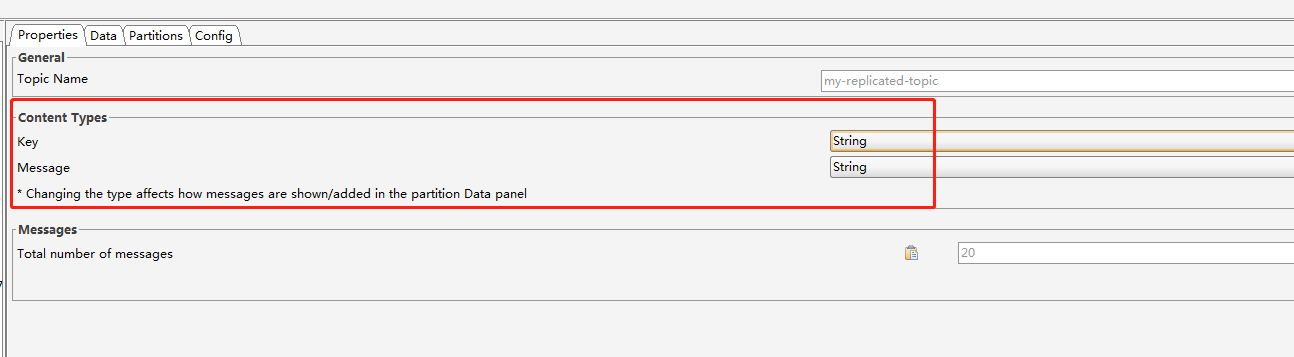


消息的key、value都不是必填的。消息类型是Byte Array的时候必须是16进制的串，写入消息的时候也需要将消息转换成16进制的。不太直观，可以修改成string，会方便很多。

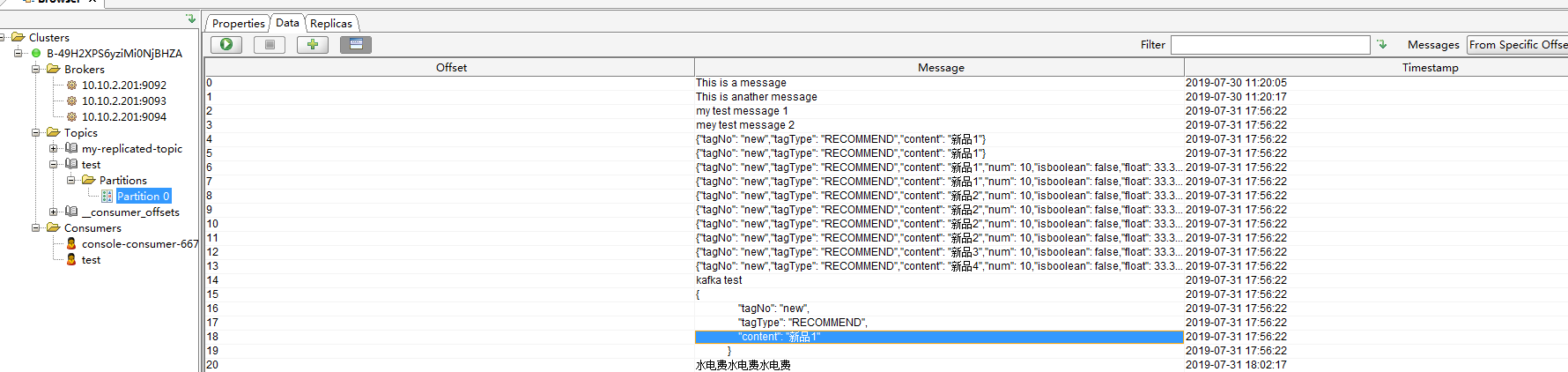
查看消费者的信息



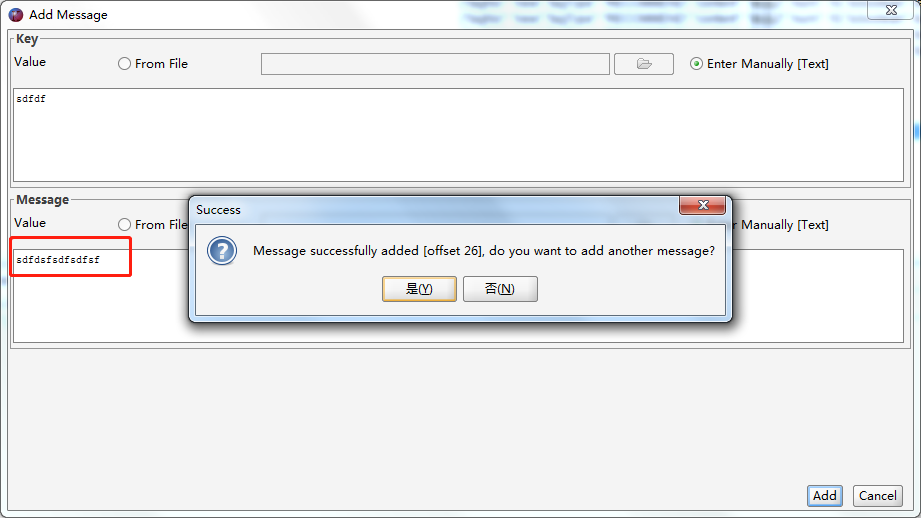
这个消息类型改成string后



消息内容直观很多，方便很多



添加消息的时候也不需要转换成16进制



问题：kafka tool通过文件的方式添加消息，好像一个文件只能添加一条消息。

能导入文件可以添加多条消息就好了。