KIP-641 An new java interface to replace 'kafka.common. MessageReader'

This page is meant as a template for writing a KIP. To create a KIP choose Tools->Copy on this page and modify with your content and replace the heading with the next KIP number and a description of your issue. Replace anything in italics with your own description.

Status

Current state: *adopted*

Discussion thread: *here*

VOTE: *here*

JIRA: *here*

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

kafka.common.MessageReader is a input argument of kafka-console-producer and we expect users can have their custom reader to produce custom records. Hence, *MessageReader* is a public interface and we should offer a java version to replace current scala code. Also, the new *MessageReader* shou ld be placed at clients module. (*kafka.common.MessageReader* is in core module)

Public Interfaces

New org.apache.kafka.tools.api.RecordReader interface and new module tools-api

```
/**
 * Typical implementations of this interface convert data from an `InputStream` received via `readRecords` into
a
 * iterator of `ProducerRecord` instance. Noted that the implementations to have a public nullary constructor.
 *
 * This is used by the `kafka.tools.ConsoleProducer`.
 */
public interface RecordReader extends Closeable, Configurable {
    /**
     * read byte array from input stream and then generate a iterator of producer record
     * @param inputStream of message. the implementation does not need to close the input stream.
     * @return a iterator of producer record. It should implement following rules. 1) the hasNext() method must
be idempotent.
     *         2) the convert error should be thrown by next() method.
     */
    Iterator<ProducerRecord<byte[], byte[]>> readRecords(InputStream);


    /**
     * Closes this reader. This method is invoked if the iterator from readRecords either has no more records
or throws exception.
     */
    default void close() {}
}
```

Proposed Changes

1. Deprecate kafka.common.MessageReader

```
@deprecated("This class has been deprecated and will be removed in 4.0. Please use org.apache.kafka.tools.api.
RecordReader instead", "3.5.0")
trait MessageReader
```

2. log warning messages to users when using deprecated MessageReader

Compatibility, Deprecation, and Migration Plan

1. backward compatibility
   kafka.common.MessageReader implementations can keep working without recompilation.
2. deprecation
   a. kafka.common.MessageReader is deprecated
   b. the method init(InputStream inputStream, Properties props) is deprecated and replacement is configure(Map<String, ?> configs)
3. migration plan: users have address following changes to complete code migration
   a. import the new dependency: tools-api
   b. change inheritance from kafka.common.MessageReader to org.apache.kafka.client.tools.api.RecordReader
   c. change method signature from init(InputStream inputStream, Properties props) to configure(Map<String, ?> configs)
   d. remove the implementation of readMessage()
   e. implements the readRecords(InputStream) to return Iterator of records

Rejected Alternatives

1. support the usage of Serializers (from Juma): this support could complicate the configs, since users have to define both MessageReader and serializer. It seems to me the mechanism of MessageReader should include serialization.
2. move RecordReader to "org.apache.kafka.common": "org.apache.kafka.common" does not allow to import code from "org.apache.kafka.clients.producer". However, the tool-related interface should be able to access producer, consumer and admin code.
3. move RecordReader to "org.apache.kafka.clients.tool": client module already has many pluggable interfaces. we should follow the package naming. The server-related pluggable interfaces are located at "org.apache.kafka.server", and thus tools-related interface should be located at "org.apache.kafka.tools"
4. ~~new RecordReader implements Configurable: Implementing Configurable will change the arguments of configure method from (InputStream, configs) to (configs). That obstructs RecordReader from keeping input stream itself as some of that state.~~
5. configurable(InputStream, configs) - diverges from the Configurable interface and it is strange to pass an InputStream to a configure method.
6. move RecordReader to "org.apache.kafka.tools" (client module): The tools module has same package so we should not create the same package on another module to avoid split package (https://www.logicbig.com/tutorials/core-java-tutorial/modules/split-packages.html)