KIP-254: JsonConverter Exception Handeling

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: *Under Discussion*

Discussion thread: *here [Change the link from the KIP proposal email archive to your own email thread]*

JIRA: *here*

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

Motivation

If we use JSONConvertor to serialize and deserialize the message, any one malformed JSON message that is in the input Kafka topic will stop the functioning of the client / producer / connector / streams which is using this convertor. We need to provide user with the option to specify how they can handle this issue. They can either specify if they would like to skip this message and move on or fail the process that is using this convertor. Failing of the process is currently the default with this change the user can specify if they are fine with skipping the malformed JSON message. This can however introduce a dataloss when the malformed message is valid and are corrupted in flight.

Public Interfaces

Briefly list any new interfaces that will be introduced as part of this proposal or any existing interfaces that will be removed or changed. The purpose of this section is to concisely call out the public contract that will come along with this feature.

A public interface is any change to the following:

- *Binary log format*
- *The network protocol and api behavior*
- Any class in the public packages under clientsConfiguration, especially client configuration
    - org/apache/kafka/common/serialization
    - org/apache/kafka/common
    - org/apache/kafka/common/errors
    - org/apache/kafka/clients/producer
    - org/apache/kafka/clients/consumer (eventually, once stable)
- *Monitoring*
- *Command line tools and arguments*
- *Anything else that will likely break existing users in some way when they upgrade*

Proposed Changes

Describe the new thing you want to do in appropriate detail. This may be fairly extensive and have large subsections of its own. Or it may be a few sentences. Use judgement based on the scope of the change.

Compatibility, Deprecation, and Migration Plan

- *What impact (if any) will there be on existing users?*
- *If we are changing behavior how will we phase out the older behavior?*
- *If we need special migration tools, describe them here.*
- *When will we remove the existing behavior?*

Rejected Alternatives

If there are alternative ways of accomplishing the same thing, what were they? The purpose of this section is to motivate why the design is the way it is and not some other way.