KIP-DRAFT: Handle errors for data inside put method of a sink connector

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: TBD

JIRA: Unable to render Jira issues macro, execution error.

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

Motivation

Kafka connectors which interact with database specially sink connectors need to know how to handle field length mismatches. Most databases like oracle enforce field lengths but there is no way to enforce the same on Avro.

Sample applies with not null fields where records coming in with empty strings need to be rejected.

We probably can write KSQL or stream jobs to filter out these records but the KSQL query can get very big and difficult to manage as there might be hundreds of fields in a table.

Also http sink connector needs to send back messages which could not be handled to DLQ.

An easier approach probably will be in the connect put method to filter out these records using the database metadata already available to the connector and then either discard these bad records or write them into a DLQ topic.

Public Interfaces

This KIP is probably an extension of KIP-298. Particularly the DLQ part which will be used by the connector to send bad records.

These are the DLQ related API changes mentioned in KIP-298.

Config Option	Description	Default Value	Domain
errors.deadletterqueue.context.headers.enable	If true, multiple headers will be added to annotate the record with the error context	false	Boolean
errors.deadletterqueue.topic.name	The name of the dead letter queue topic. If not set, this feature will be disabled.	""	A valid Kafka topic name
errors.deadletterqueue.topic.replication.factor	Replication factor used to create the dead letter queue topic when it doesn't already exist.	3	[1 ... Short.MAX_VALUE]

Proposed Changes

The put method inside the connector task should support passing the DLQ producer to it.

The writer should read the cached table definition to get the column definitions.

Then it should scan through all records and figure out the records which violate table definition (length and not null check).

It should commit the clean records and send back the violation list back the task put method which it can use to write into the DLQ topic.

It can write the violation cause as a header in the message.

An example of how it can be done is in this git repo. Currently its extending the jdbc connector and its adding a set of extra parameters to retrieve the DLQ details.

Compatibility, Deprecation, and Migration Plan

- As mentioned it might be dependent on KIP-298 for the DLQ part
- If DLQ is not set and we just have to ignore the messages which do not match the DB field lengths then it can be done without KIP-298.

Rejected Alternatives

As mentioned we can write stream/KSQL queries to filter out records which do not match DB lengths but they will need to be done for all fields which might end up being a very large task to manage and very unstable as well, as any change in the database without corresponding change in the KSQL will bring down the connector.