

KIP-135 : Send of null key to a compacted topic should throw non-retriable error back to user

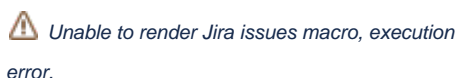
- [Status](#)
- [Motivation](#)
- [New or Changed Public Interfaces](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

Status

Current state: *"Under Discussion"*

Discussion thread: [here](#)

JIRA:

A screenshot of a JIRA error message. It features a yellow warning triangle icon on the left, followed by the text "Unable to render Jira issues macro, execution error." in a standard sans-serif font.

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

Motivation

Currently when a user sends a null key to a log compacted topic, the broker returns `CorruptRecordException`, which is a retrieable exception. As such, the producer keeps retrying until retries are exhausted or `request.timeout.ms` expires and eventually throws a `TimeoutException`. This is confusing and not user-friendly.

The broker should throw a non-retriable exception with a message explaining the error.

New or Changed Public Interfaces

This KIP introduces a new error type `INVALID_KEY`.

It introduces a new non-retriable exception `InvalidRecordKeyException`. Going forward user request will fail without doing retries, if they try to produce to a log compacted topic with a null key.

Proposed Changes

Add a new error code and exception as follows :

Error Code :

```
INVALID_RECORD_KEY(45, new InvalidRecordKeyException("Message key is invalid.))
```

Exception :

```
public class InvalidRecordKeyException extends ApiException {  
  
    private static final long serialVersionUID = 1L;  
  
    public InvalidRecordKeyException() {  
        super();  
    }  
  
    public InvalidRecordKeyException(String message, Throwable cause) {  
        super(message, cause);  
    }  
  
    public InvalidRecordKeyException(String message) {  
        super(message);  
    }  
  
    public InvalidRecordKeyException(Throwable cause) {  
        super(cause);  
    }  
}
```

Compatibility, Deprecation, and Migration Plan

We will only return this exception for ProduceRequest V3 or higher. The older version of ProduceRequest will still get the `CorruptRecordException`.

Rejected Alternatives