

# KIP-523: Add KStream#toTable to the Streams DSL


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


## Status

**Current state:** *Accepted* [Vote](#)

**Discussion thread:** <https://lists.apache.org/thread.html/a5ee2d569448dd57647dca93f288405c13785dec9cb4a0cbddfedd35@%3Cdev.kafka.apache.org%3E>

JIRA:

 Unable to render Jira issues macro, execution error.

 Unable to render Jira issues macro, execution error.

## Motivation

For users who cannot read their source topic as a changelog stream from the beginning we need to provide a way for event streams to be translated into changelog stream. As pointed out by [guozhang Wang](#) this should not be confused with `KStream.reduce()`, these functions should:

1. completely change the semantics of the input stream from event stream to changelog stream any null-values will still be serialized, and if the resulted bytes are also null they will be interpreted as "deletes" to the materialized KTable (i.e. tombstones in the changelog stream).
2. Materialization of the KTable will follow the usual process:
  - a. if Optimization is turned off, the KTable will always be materialized (but the store will not be queryable)
  - b. if Optimization is turned on and if `.toTable()` is used, the Ktable may or may not be materialized. The store still cannot be queried.
  - c. if `.toTable(final Materialized<K, V, KeyValueStore<Bytes, byte[]>> materialized)` is used irrespective of the optimization strategy used the KTable will be materialized and will be queryable.

## Public Interfaces

Two functions will need to be added to the KStream interface:

```
//java
KTable<K,V> toTable()
KTable<K,V> toTable(Named)
KTable<K,V> toTable(final Materialized<K, VR, KeyValueStore<Bytes, byte[]>> materialized)
KTable<K,V> toTable(Named, final Materialized<K, VR, KeyValueStore<Bytes, byte[]>> materialized)

//scala
def toTable: KTable[K, V]
def toTable(materialized: Materialized[K, V, ByteArrayKeyValueStore]): KTable[K, V]
```

## Proposed Changes

Adding two new functions to the Kstreams DSL.

1. When naming the processor & state stores the same naming convention needs to be followed as specified in [KIP-307: Allow to define custom processor names with KStreams DSL](#)

## Compatibility, Deprecation, and Migration Plan

*No users will be impacted as this is just addition of two new methods.*

## Rejected Alternatives

*None.*