

KIP-420: Add Single Value Fetch in Session Stores

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

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 (2.2.0)*

Discussion thread: [link](#)

JIRA: NA

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

Motivation

When fixing the issue in

 Unable to render Jira issues macro, execution error.

regarding the wrong old values being flushed to

downstream, I realized the root cause was that we are trying to fetch the exact session's key with the provided key, start-time, end-time, but since there's no single-fetch API we are mimicking it with a range buggy query. On the other hand, like WindowStore, if users know which session they are querying exactly, they should be able to issue a "single-point" query, which should be much less costly as well compared to a range query. Such API could also help fixing the immediate issue of KAFKA-7652 as well.

Public Interfaces

This KIP would propose to add a single fetch API to the SessionStore interface:

```
AGG fetchSession(K key, long startTime, long endTime);
```

Internally, this function will be used in 1) caching store flush listener, to read the old value from underlying store; 2) for accessing values from the session store in storeGetter.

Proposed Changes

As above.

Compatibility, Deprecation, and Migration Plan

- *Users implementing customized session-store would now need to make code changes to add one more function implementation for the added API. The reason that we've decided to not have an default implementation (e.g. have a very conservative range query, and use a condition to filter out all but only leave one that has the exact start/end timestamp) in the public API is that this one is going to be used inside the library as well at critical code paths, e.g. caching store's flush logic. And hence we really need this API implementation to be efficient.*

Rejected Alternatives

1. *I've considered adding this API to ReadOnlySessionStore, but realized it is not a very useful API for interactive queries since most IQ users would not know the exact session start-end timestamps.*