

# KIP 155 - Add range scan for windowed state stores

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

## Status

**Current state:** *Accepted*

**Discussion thread:** [here](#)

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

Unlike for key-value state stores, Kafka Streams currently does not provide a way to query the range of keys available in a windowed state store.

The only available alternative is to aggregate all the distinct keys into a single key for a given window, or to implement your own windowed state store.

Similarly, session stores currently only offer querying by a given key, requiring the user to maintain the set of queryable keys separately.

## Public Interfaces

This KIP would add the following methods:

`ReadOnlyWindowStore` interface

```
KeyValueIterator<Windowed<K>, V>> fetch(K from, K to, long timeFrom, long timeTo)
```

The time range would follow the existing `ReadOnlyWindowStore.fetch(K key, long timeFrom, long timeTo)` behavior for the time range.

Key range behavior would be consistent with the existing `ReadOnlyKeyValueStore.range(K from, K to)` behavior.

`ReadOnlySessionStore` interface

```
KeyValueIterator<Windowed<K>, AGG> fetch(final K from, final K to)
```

Key range behavior would be consistent with the existing `ReadOnlyKeyValueStore.range(K from, K to)` behavior.

`SessionStore` interface

```
KeyValueIterator<Windowed<K>, AGG> findSessions(final K keyFrom, final K keyTo, long earliestSessionEndTime, final long latestSessionStartTime);
```

Key range behavior would be consistent with the existing `ReadOnlyKeyValueStore.range(K from, K to)` behavior.

## Proposed Changes

This KIP proposes to add the interface described above and implement range scan returning all the entries in the given key range.

## Compatibility, Deprecation, and Migration Plan

- Users implementing their own state stores would be affected by the interface changes.

## Rejected Alternatives

The existing return type in `ReadOnlyWindowStore.fetch((K key, long timeFrom, long timeTo)` is not ideal, since `WindowStoreIterator` is a `KeyValueIterator<Long, V>`, which abuses the key as a timestamp and the value as the object of interest. However we want to be able to return the keys as part of range scans, so we considered using `WindowStoreIterator<KeyValue<K, V>>` as the return type for windowed stores for consistency between single key and range scan methods.

It was pointed out that doing so would limit the usefulness of `peekNextKey()` on the iterator, while also being somewhat confusing, since the iterator key doesn't actually contain the keys. As a result, it seemed simpler to follow the model already used in `ReadOnlySessionStore` to return use a `KeyValueIterator` with a `Windowed<K>` as the key.