

KIP-396: Add Reset/List Offsets Operations to AdminClient

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

Status

Current state: Accepted

Discussion thread: [here](#)

JIRA: [KAFKA-7689](#)

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

Motivation

Resetting the offsets of a consumer group is a relatively common operation. It's usually done to allow an application to reprocess data or otherwise to skip/drop records. Currently (thanks to [KIP-122](#)) it can be done using the `kafka-consumer-group.sh` tool however it would be nice to be able to do it directly with the `AdminClient`. Applications can more easily depend on the `AdminClient` API rather than running the `ConsumerGroupsCommand` class or worse reimplementing the logic. The ability to retrieve committed offsets is already available in the `AdminClient` and has proved very useful. Additionally [KIP-496](#) is adding the ability to delete committed offsets. This justifies integrating the consumer group reset/alter functionality directly in the `AdminClient`.

Public Interfaces

Plan is to add 4 new methods to the `AdminClient` public API:

AdminClient.java

```
/**
 * <p>Alters offsets for the specified group.
 *
 * <p>This is a convenience method for {@link #alterConsumerGroupOffsets(String, Map, AlterOffsetsOptions)}
with default options.
 * See the overload for more details.
 *
 * @param groupId The group for which to alter offsets.
 * @param offsets A map of offsets by partition with associated metadata.
 * @return The AlterOffsetsResult.
 */
public AlterOffsetsResult alterConsumerGroupOffsets(String groupId, Map<TopicPartition, OffsetAndMetadata>
offsets) {
    return alterOffsets(groupId, offsets, new AlterOffsetsOptions());
}

/**
 * <p>Alters offsets for the specified group.
 *
 * <p>This operation is not transactional so it may succeed for some partitions while fail for others.
 *
 * @param groupId The group for which to alter offsets.
 * @param offsets A map of offsets by partition with associated metadata. Partitions not specified in the
map are ignored.
 * @param options The options to use when altering the offsets.
 * @return The AlterOffsetsResult.
 */
public abstract AlterOffsetsResult alterConsumerGroupOffsets(String groupId, Map<TopicPartition,
OffsetAndMetadata> offsets, AlterOffsetsOptions options);

/**
 * <p>List offset for the specified partitions.
 *
 * <p>This is a convenience method for {@link #listOffsets(Collection, ListOffsetsOptions)}
 *
 * @param topicPartitionOffsets The mapping from partition to the OffsetSpec to look up.
 * @return The ListOffsetsResult.
 */
public ListOffsetsResult listOffsets(Map<TopicPartition, OffsetSpec> topicPartitionOffsets) {
    return listOffsets(topicPartitionOffsets, new ListOffsetsOptions());
}

/**
 * <p>List offset for the specified partitions.
 *
 * @param topicPartitionOffsets The mapping from partition to the OffsetSpec to look up.
 * @param options The options to use when retrieving the offsets
 * @return The ListOffsetsResult.
 */
public abstract ListOffsetsResult listOffsets(Map<TopicPartition, OffsetSpec> topicPartitionOffsets,
ListOffsetsOptions options);
```

We will also add companion objects under `org.apache.kafka.clients.admin` for these methods:

- `AlterOffsetsResult`:

AlterOffsetsResult.java

```
/**
 * The result of the {@link AdminClient#alterOffsets(String, Map)} call.
 *
 * The API of this class is evolving, see {@link AdminClient} for details.
 */
@InterfaceStability.Evolving
public class AlterOffsetsResult {

    public AlterOffsetsResult(KafkaFuture<Map<TopicPartition, KafkaFuture<Void>>> futures) {}

    public KafkaFuture<Map<TopicPartition, KafkaFuture<Void>>> values() {}

    public KafkaFuture<Map<TopicPartition, Void>> all() {}
}
```

- AlterOffsetsOptions

AlterOffsetsOptions.java

```
/**
 * Options for the {@link AdminClient#alterOffsets()} call.
 *
 * The API of this class is evolving, see {@link AdminClient} for details.
 */
@InterfaceStability.Evolving
public class AlterOffsetsOptions extends AbstractOptions<AlterOffsetsOptions> {
}
```

- ListOffsetsResult

ListOffsetsResult.java

```
/**
 * The result of the {@link AdminClient#listOffsets(String)} call.
 *
 * The API of this class is evolving, see {@link AdminClient} for details.
 */
@InterfaceStability.Evolving
public class ListOffsetsResult {

    public ListOffsetsResult(Map<TopicPartition, KafkaFuture<ListOffsetsResultInfo>> futures) {}

    /**
     * Return a map from TopicPartition to futures which can be used to retrieve the offsets
     */
    public Map<TopicPartition, KafkaFuture<ListOffsetsResultInfo>> values() {}

    /**
     * Return a future which succeeds only if offsets for all specified partitions have been successfully
     * retrieved.
     */
    public KafkaFuture<Map<TopicPartition, ListOffsetsResultInfo>> all() {}

    static public class ListOffsetsResultInfo {

        ListOffsetsResultInfo(long offset, long timestamp, Optional<Integer> leaderEpoch) {}

        long offset() {}
        long timestamp() {}
        Optional<Integer> leaderEpoch() {}
    }
}
```

- ListOffsetsOptions

ListOffsetsOptions.java

```
/**
 * Options for {@link AdminClient#listOffsets()}.
 *
 * The API of this class is evolving, see {@link AdminClient} for details.
 */
@InterfaceStability.Evolving
public class ListOffsetsOptions extends AbstractOptions<ListOffsetsOptions> {

    private final IsolationLevel isolationLevel;

    // isolationLevel default to READ_UNCOMMITTED
    public ListOffsetsOptions() {}
    public ListOffsetsOptions(IsolationLevel isolationLevel) {}

    public IsolationLevel isolationLevel() {}
}
```

- OffsetSpec

OffsetSpec.java

```
/**
 * This class allows to specify the desired query when retrieving offsets
 */
public class OffsetSpec {

    public OffsetSpec latest() {}

    public OffsetSpec earliest() {}

    public OffsetSpec forTimestamp(long timestamp) {}
}
```

- IsolationLevel (The existing IsolationLevel class is not part of the public API, hence we are creating this new file. The old class could then be removed)

IsolationLevel.java

```
/**
 * The isolation level
 */
public enum IsolationLevel {
    READ_UNCOMMITTED((byte) 0),
    READ_COMMITTED((byte) 1);
}
```

Proposed Changes

The proposal of this KIP is to add methods to alter offsets for a group in the `AdminClient`. Alongside, methods to retrieve log offsets (ListOffset API) so users can easily retrieve start and end offsets or offsets for a timestamp for partitions will be added.

The 4 new methods in `AdminClient`:

- `public AlterOffsetsResult alterConsumerGroupOffsets(String groupId, Map<TopicPartition, OffsetAndMetadata> offsets) {}`
- `public abstract AlterOffsetsResult alterConsumerGroupOffsets(String groupId, Map<TopicPartition, OffsetAndMetadata> offsets, AlterOffsetsOptions options);`
- `public ListOffsetsResult listOffsets(Map<TopicPartition, OffsetSpec> topicPartitionOffsets) {}`
- `public abstract ListOffsetsResult listOffsets(Map<TopicPartition, OffsetSpec> topicPartitionOffsets, ListOffsetsOptions options);`

Regarding the implementation, instead of relying on `KafkaConsumer` to perform the alter, I propose to instead directly send `OffsetCommit` requests. This is principally for 2 reasons

- Be more performant. In order to alter offsets, a `KafkaConsumer` has to be created for each group. It first needs to subscribe or be assigned topic /partitions before being able to alter offsets. Additionally we've need to pause consumption so calls to poll to perform the alter operation don't return records. That would require more connections/disconnections to the cluster, more network requests.
- Avoid having the `AdminClient` depend on `KafkaConsumer`. Currently `AdminClient` does not depend on another client.

Compatibility, Deprecation, and Migration Plan

No impact or migration plan required as this proposal is only adding new methods and not changing any current behaviour.

Rejected Alternatives

- *Implement offset reset like it is currently done in `ConsumerGroupCommand`. As mentioned in the motivation section, this is inefficient.*