

KIP-222 - Add Consumer Group operations to Admin API

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

Status

Current state: "Accepted"

Discussion thread: [here](#)

JIRA: [here](#)

 Unable to render Jira issues macro, execution error.

Released: 2.0.0

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

Motivation

This KIP is aim to add support for describing consumer groups and list consumer groups to `KafkaAdminClient` class. This functionality is required on the S

treams Resetter Tool as describe here:

 Unable to render Jira issues macro, execution error.

Public Interfaces

API

Add the following API in AdminClient:

```
public abstract class AdminClient implements AutoCloseable {

    public abstract DescribeConsumerGroupsResult describeConsumerGroups(Collection<String>
groupId, DescribeConsumerGroupsOptions options);

    public DescribeConsumerGroupsResult describeConsumerGroups(Collection<String> groupId) {
        return describeConsumerGroups(groupId, new DescribeConsumerGroupsOptions());
    }

    public abstract ListConsumerGroupsResult listConsumerGroups(ListConsumerGroupsOptions options);

    public ListConsumerGroupsResult listConsumerGroups() {
        return listConsumerGroups(new ListConsumerGroupsOptions());
    }
}
```

```

    }

    public abstract ListConsumerGroupOffsetsResult listConsumerGroupOffsets(String groupId,
ListConsumerGroupsOptions options);

    public ListConsumerGroupOffsetsResult listConsumerGroupOffsets(String groupId) {
        return listConsumerGroupOffsets(new ListConsumerGroupOffsetsOptions());
    }

    public abstract DeleteConsumerGroupsResult deleteConsumerGroups(Collection<String> groupIds,
DeleteConsumerGroupsOptions options);

    public DeleteConsumerGroupsResult deleteConsumerGroups(Collection<String> groupIds) {
        return deleteConsumerGroups(groupIds, new DeleteConsumerGroupsOptions());
    }
}

public class DescribeConsumerGroupOptions extends AbstractOptions<DescribeConsumerGroupOptions> {
}

public class DescribeConsumerGroupResult {
    Map<TopicPartition, KafkaFuture<ConsumerGroupDescription>> all();
}

public class ConsumerGroupDescription {
    /**
     * The id of the consumer group.
     */
    public String groupId();

    /**
     * If consumer group is simple or not.
     */
    public boolean isSimpleConsumerGroup();

    /**
     * A list of the members of the consumer group.
     */
    public Collection<MemberDescription> members();

    /**
     * The consumer group partition assignor.
     */
    public String partitionAssignor();

    /**
     * The consumer group state, or UNKNOWN if the state is too new for us to parse.
     */
    public ConsumerGroupState state();

    /**
     * The consumer group coordinator, or null if the coordinator is not known.
     */
    public Node coordinator();
}

public enum ConsumerGroupState {
    UNKNOWN,
    PREPARING_REBALANCE,
    COMPLETING_REBALANCE,
    STABLE,
    DEAD,
    EMPTY;
}

public class MemberDescription {
    /**
     * The consumer id of the group member.

```

```

    */
    public String consumerId();

    /**
     * The client id of the group member.
     */
    public String clientId();

    /**
     * The host where the group member is running.
     */
    public String host();

    /**
     * The assignment of the group member.
     */
    public MemberAssignment assignment();
}

public class MemberAssignment {
    private final List<TopicPartition> topicPartitions;
}

public class ListConsumerGroupsOptions extends AbstractOptions<ListConsumerGroupsOptions> {
}

public class ListConsumerGroupsResult {
    public KafkaFuture<Collection<ConsumerGroupListing>> all();
    public KafkaFuture<Collection<ConsumerGroupListing>> valid();
    public KafkaFuture<Collection<Throwable>> errors();
}

public class ConsumerGroupListing {
    private final String groupId;
    private final boolean isSimpleConsumerGroup;
}

public class ListConsumerGroupOffsetsOptions extends AbstractOptions<ListConsumerGroupsOptions> {
}

public class ListConsumerGroupOffsetsResult {
    KafkaFuture<Map<TopicPartition, OffsetAndMetadata>> all();
}

public class DeleteConsumerGroupOptions extends AbstractOptions<DeleteConsumerGroupOptions> {
}

public class DeleteConsumerGroupsResult {
    Map<TopicPartition, KafkaFuture<Void>> all();
}

```

This API returns a future object whose result will be available within RequestTimeoutMs, which is configured when user constructs the AdminClient.

Proposed Changes

1. Add ``#describeConsumerGroups(Collection<String> groupIds, DescribeConsumerGroupsOptions options)`` and ``#describeConsumerGroups(Collection<String> groupIds)`` to ``AdminClient`` API
2. Add ``#listConsumerGroups(ListConsumerGroupsOptions options)`` and ``#listConsumerGroups()`` to ``AdminClient`` API
3. Add ``#listConsumerGroupOffsets(String groupId, ListConsumerGroupOffsetsOptions options)`` and ``#listConsumerGroupOffsets(String groupId)`` to ``AdminClient`` API and
4. Implement it on ``KafkaAdminClient``.

5. Solve

 Unable to render Jira issues macro, execution error.

6. Add ``#deleteConsumerGroups(Collection<String> groupIds, ConsumerGroupsOptions options)`` and ``#deleteConsumerGroups(Collection<String> groupIds)`` to ``AdminClient`` API based on [KIP-229: DeleteGroups API](#).

Compatibility, Deprecation, and Migration Plan

- There won't be any impact on existing users.
- There won't be any change of current behavior.
- No migration tool required.

Rejected Alternatives

- We are moving one more functionality from `core`'s `AdminClient` API to `clients`'s `AdminClient` API. This is aim to remove dependencies to `core` module.
- Aggregating both Consumer Groups and Connect Groups in the same API would increase complexity adding generic MemberAssignment to match both groups. This KIP is only including Consumer Groups and will leave Connect Groups to be handled by another KIP. <https://github.com/apache/kafka/pull/4454#issuecomment-360553277>

Future Work

- Streams Resetter Tool will be remove it's dependency to `core` module for `KafkaAdminClient`.

References:

Pull Request: <https://github.com/apache/kafka/pull/4454>