

KIP-327: Add describe all topics API to AdminClient

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

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: *Discarded*

Discussion thread: [here](#)

JIRA: [here](#)

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

Motivation

Currently AdminClient supports describeTopics(Collection<String> topicNames) API for describing topics and listTopics() API for listing topic names. To describe all topics, we need to use listTopics() to get all topic names and supply the name list to describeTopics.

In large clusters, the MetadataResponse can be large and sending multiple MetadataRequests can be inefficient.

Since "describe all topics" is a common operation, We propose to add describeTopics() API to get all topic descriptions in single call. This will be simple to use and avoids additional metadata requests.

Public Interfaces

API Changes

We will introduce new API methods to AdminClient to describe all topics.

AdminClient.java

```
/**
 * Describe all topics in the cluster, with the default options.
 */
public DescribeAllTopicsResult describeTopics() {
    return describeTopics(new DescribeTopicsOptions());
}

/**
 * Describe all topics in the cluster.
 *
 * @return The DescribeAllTopicsResult.
 */
public abstract DescribeAllTopicsResult describeTopics(DescribeTopicsOptions options);
```

New result class "DescribeAllTopicsResult" will be added.

DescribeAllTopicsResult

```
/**
 * The result of the {@link KafkaAdminClient#describeTopics()} call.
 */
public class DescribeAllTopicsResult {
    private final KafkaFuture<Collection<TopicDescription>> future;

    DescribeAllTopicsResult(KafkaFuture<Collection<TopicDescription>> future) {
        this.future = future;
    }

    public KafkaFuture<Collection<TopicDescription>> value() {
        return future;
    }
}
```

Compatibility, Deprecation, and Migration Plan

- There won't be any compatibility issues