KIP-533: Add default api timeout to AdminClient

Status

Current state: Adopted

Discussion thread:

JIRA: **KAFKA-8503** - Getting issue details...   STATUS

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

Motivation

Currently retry/timeout behavior in the AdminClient is difficult to control. We can limit the total number of retries and we can control the individual request timeout. The problem is that it is difficult for users to reason about the admin client at the request level. Some requests may be retried in quick succession; others may reach the request timeout before retrying. This makes it hard to bound the total time for an operation to complete which is really what most users care about.

This problem is partially addressed by allowing users to pass through a timeout in the Options argument, but the gap is having a way to tune this timeout through configuration. Today, we overload the request timeout for this purpose: if no explicit api timeout has been provided, then we will use the request timeout. However, this means we need a large default request timeout in order to have sensible default timeout behavior. A further problem is that the default max number of retries is currently 5, which means we often have requests failing prior to the timeout being exceeded.

This proposal aims to address these problems by introducing a separate api timeout configuration which is decoupled from the request timeout. We also propose to adjust the default configurations for more reasonable out-of-the-box experience.

Public Interfaces

We will add a new configuration `default.api.timeout.ms` to the AdminClient. We will also update the default values for `retries` and `request.timeout.ms`. The table below summarizes the changes.

Configuration	Current default	New default
default.api.timeout.ms	N/A	60000
request.timeout.ms	120000	30000
retries	5	Int.MaxValue

Proposed Changes

We will introduce a new `default.api.timeout.ms` (named for consistency with a similar config in the consumer). If no explicit timeout has been provided in the Options argument for a given method, this value will be used. We will also adjust the default number of retries to be effectively unlimited. We will preserve the current behavior of failing an operation when either the timeout or the maximum number of retries has been reached.

We also change the default request timeout since it is now decoupled from the api timeout. Note that each admin API allows for an override of the api timeout through its Options class (e.g. `CreateTopicsOptions`). We will not initially allow request timeout overrides, though this remains a potential improvement for the future.

Compatibility, Deprecation, and Migration Plan

Changing the default values always carries some compatibility risk, but we believe it is worth paying the cost for a better default experience. The risk in any case should be low since the defaults remain fairly conservative.

Rejected Alternatives

We have considered deprecating the `retries` configuration, but ultimately decided to revisit this in the future once we are more confident it is no longer needed.