

KIP-774: Deprecate public access to Admin client's *Result constructors

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

Status

Current state: *Under Discussion*

Discussion thread: [here](#)

JIRA: [KAFKA-13285](#)

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

Motivation

The `*Result` classes returned by the `Admin` client originally had package access constructors. This allowed these constructors to be changed without needing to reason about compatibility, since client code could not directly instantiate these classes.

Over time a minority of the `Result` classes have ended up with public constructors. So there is an inconsistency in these classes. This [caused a problem](#) in the Kafka 3.0 development cycle because a couple of constructor signatures were changed, but one of those constructors was public, which was missed in the original code review.

So to have a consistent API and avoid similar problems in the future the inconsistency ought to be resolved, either by making all the constructors public or or all package access.

[KIP-692](#) proposed to make all these constructors public, using the rationale that it simplifies the creation of mock `Admin` clients for testing. Note that when the vote for KIP-692 was called it was done on the [DISCUSS] thread, so the lack of votes might not be an sure indication of how the community feels about it.

This KIP proposes taking a step in the opposite direction: To deprecate the public use of these constructors and eventually to make them package access.

Trade off is:

- While the creation of `Admin` mocks with package constructors is not `_ergonomic_`, it is `_possible_`. The example code in KIP-692 requires two line of codes for each result instance.
- Having public constructors can hinder the evolution of the `Admin` client. A number of these constructors have changed their signature during the life of the `Admin` client. Further change can never be discounted. Sometimes these changes have been non-trivial. For example, while it's easy to overload a constructor (e.g. to add an extra future in a compatible way), overloading can't be used for making compatible changes when a generic type argument needs to be changed. Since we can't anticipate how the `Admin Result` classes might need to evolve in the future it's simpler and safer to avoid exposing these constructors than it is to need to reason about making source and binary compatible changes.

Public Interfaces

Deprecate the public use of the constructors of the following classes:

- `AlterClientQuotasResult`
- `AlterUserScramCredentialsResult`
- `DeleteRecordsResult`
- `DescribeClientQuotasResult`
- `DescribeConsumerGroupsResult`
- `ListOffsetsResult`

Also, remove the `final` modifier from `ElectLeadersResult`. This prevents the instantiation of mocks by frameworks such as Mockito, and appear to confer no safety on the use of the result class.

Proposed Changes

Per public interfaces section.

Compatibility, Deprecation, and Migration Plan

The constructors will become package access in a future major release. Existing out-of-package callers will need to use mock libraries or reflection to instantiate these classes. The only anticipated existing out-of-package callers are in other project's test code.

Rejected Alternatives

- Do nothing.
- Achieve consistency by making the constructors public, as proposed by KIP-692, and deal with compatibility problems on a KIP-by-KIP- basis.
- Provide some other public API for the creation of instances of these classes to address the verbosity of mock creation. Providing a supportable API in the face of changing underlying constructors seems like a lot of work for only marginal benefit.
- Change the Result classes to interfaces. This would be quite a bit more code, for similar protection as a package-private constructor and not improve the situation for people mocking the Admin client.