

KIP-621: Deprecate and replace DescribeLogDirsResult.all() and .values()

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

Status

Current state: *Accepted*

Discussion thread: [here](#)

JIRA:

 Unable to render Jira issues macro, execution error.

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

Motivation

Classes in `org.apache.kafka.common.requests` are not considered part of the supported public API. Unfortunately the `DescribeLogDirsResponse` e. `LogDirInfo` class from this package is used as the declared return type of the `all()` and `values()` methods of `DescribeLogDirsResult`, which *is* in the public API. Furthermore the `DescribeLogDirsResponse.ReplicaInfo` and `Errors` classes are reachable from the public API of `LogDirInfo`.

This means that clients are exposed to classes which are supposed to be an implementation detail and Javadoc for these classes is not published (e.g. see <https://kafka.apache.org/25/javadoc/org/apache/kafka/clients/admin/DescribeLogDirsResult.html>).

To address this new types (in `org.apache.kafka.clients.admin`) need to be created to represent the same information and either:

- use the `LogDirInfo` replacement in the return type of two *new* methods, deprecating the existing methods and eventually removing the deprecated methods in some future Kafka X.0 release, or
- change the return type of the two existing methods in some future Kafka X.0 release. In other words break binary and source compatibility and give users no notice of the change until they upgrade to X.0.

The former is preferable from a compatibility point of view, but ultimately results in method names which are inconsistent with the rest of the `*Result` classes in the Admin client (which are otherwise quite consistent in their use of `all()` and `values()`).

It should be noted that `DescribeLogDirsResult` is annotated with `@InterfaceStability.Evolving` and described in Admin as follows:

This client was introduced in 0.11.0.0 and the API is still evolving. We will try to evolve the API in a compatible manner, but we reserve the right to make breaking changes in minor releases, if necessary. We will update the `InterfaceStability` annotation and this notice once the API is considered stable.

That warning notwithstanding, this KIP proposes to follow the former route, preferring client compatibility.

Public Interfaces

The existing `DescribeLogDirsResult` methods `values()` and `all()` will be deprecated and `DescribeLogDirsResult` will get two new methods:

```

public class DescribeLogDirsResult {
    // ...

    // Deprecated:

    /**
     * Return a map from brokerId to future which can be used to check the information of partitions on each
    individual broker
     * Deprecated Since Kafka 2.7. Use {@link descriptions()}.
     */
    @Deprecated
    public Map<Integer, KafkaFuture<Map<String, LogDirInfo>>> values() {
        return futures;
    }

    /**
     * Return a future which succeeds only if all the brokers have responded without error
     * Deprecated Since Kafka 2.7. Use {@link allDescriptions()}.
     */
    @Deprecated
    public KafkaFuture<Map<Integer, Map<String, LogDirInfo>>> all() {

    // Added:

    /**
     * Return a map from brokerId to future which can be used to check the information of partitions on each
    individual broker
     */
    public Map<Integer, KafkaFuture<Map<String, LogDirDescription>>> descriptions() {
        return futures;
    }

    /**
     * Return a future which succeeds only if all the brokers have responded without error
     */
    public KafkaFuture<Map<Integer, Map<String, LogDirDescription>>> allDescriptions() {
    }
}

```

Two new classes will be added to `org.apache.kafka.clients.admin`:

```

public class LogDirDescription {
    private final Map<TopicPartition, ReplicaInfo> replicaInfos;
    private final Errors error;

    LogDirDescription(Errors error, Map<TopicPartition, ReplicaInfo> replicaInfos) {
        this.replicaInfos = replicaInfos;
    }

    /**
     * A KafkaStorageException if this log directory is offline,
     * possibly some other exception if there were problems describing the log directory
     * or null if the directory is online.
     */
    public ApiException error() {
        // ...
    }

    /**
     * A map from topic partition to replica information for that partition
     * in this log directory.
     */
    public Map<TopicPartition, ReplicaInfo> replicaInfos() {
        return unmodifiableMap(replicaInfos);
    }

    @Override
    public String toString() {
        // ...
    }
}

```

```

    }
}
public class ReplicaInfo {

    private final long size;
    private final long offsetLag;
    private final boolean isFuture;

    public ReplicaInfo(long size, long offsetLag, boolean isFuture) {
        this.size = size;
        this.offsetLag = offsetLag;
        this.isFuture = isFuture;
    }

    /**
     * The total size of the log segments in this replica in bytes.
     */
    public long size() {
        return size;
    }

    /**
     * The lag of the log's LEO with respect to the partition's
     * high watermark (if it is the current log for the partition)
     * or the current replica's LEO (if it is the {@link plain #isFuture()} future log}
     * for the partition).
     */
    public long offsetLag() {
        return offsetLag;
    }

    /**
     * Whether this replica has been created by a AlterReplicaLogDirsRequest
     * but not yet replaced the current replica on the broker.
     * @return true if this log is created by AlterReplicaLogDirsRequest and will replace the current log of
     the replica in the future.
    public boolean isFuture() {
        return isFuture;
    }

    @Override
    public String toString() {
        // ...
    }
}

```

Proposed Changes

In addition to that's described in the changes to the public interface, the `kafka.admin.LogDirsCommand` will be fixed to use the new methods.

Compatibility, Deprecation, and Migration Plan

This change is source and binary compatible.

Rejected Alternatives