KIP-45 - Standardize all client sequence interaction on j.u. Collection.

Status

Current state: "Under Discussion"

Discussion thread: *user thread*

JIRA: *KAFKA-3006*

Motivation

The KafkaConsumer API has some annoying inconsistencies in the usage of collection types. For example, subscribe() takes a list, but subscription() returns a set. Similarly for assign() and assignment(). We also have pause() , seekToBeginning(), seekToEnd(), and resume() which annoyingly use a variable argument array, which means you have to copy the result of assignment() to an array if you want to pause all assigned partitions.

Public Interfaces

Briefly list any new interfaces that will be introduced as part of this proposal or any existing interfaces that will be removed or changed. The purpose of this section is to concisely call out the public contract that will come along with this feature.

No new public interfaces are introduced, but changes to the existing ones are introduced:

```
void subscribe(Collection<String> topics);
void subscribe(Collection<String> topics, ConsumerRebalanceListener);
void assign(Collection<TopicPartition> partitions);
void pause(Collection<TopicPartition> partitions);
void resume(Collection<TopicPartition> partitions);
void seekToBeginning(Collection<TopicPartition>);
void seekToEnd(Collection<TopicPartition>);
```

The signatures available today are:

```
void subscribe(java.util.List<java.lang.String> topics);
void subscribe(java.util.List<java.lang.String> topics, ConsumerRebalanceListener listener);
void assign(java.util.List<TopicPartition> partitions);
void pause(TopicPartition... partitions);
void resume(TopicPartition... partitions);
void seekToBeginning(TopicPartition... partitions);
void seekToEnd(TopicPartition... partitions);
```

Proposed Changes

The proposed change boils down to the public interface changes above. If the first patch KAFKA-3006 is accepted, a standardization of signatures in Kafka Connect should also be considered.

Compatibility, Deprecation, and Migration Plan

- *Code that used the previous array based signatures will have to be adapted.*
- *JVMs that depend on both the 0.9.0.0 client and later versions would have incompatible signatures.*
- *These drawbacks are mitigated by the fact that KafkaConsumer is annotated with @InterfaceStability.Unstable.*
- java.util.Collection was chosen because it is a super-type of java.util.Set and java.util.List which are used as return types for **partitionsFor** and **assignment**.
- java.util.Collection as the nice added benefit of providing simple interaction with other JVM languages.

Rejected Alternatives

An alternative would be to keep the array-based versions of the calls and mark them as deprecated but this has not gathered much interest. This KIP also supersedes some of the propositions outlined in KAFKA-2991