KIP-706: Add method "Producer#send" to return CompletionStage instead of Future

Status

Current state: *Under Discussion*

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

Producer and KafkaProducer return a java.util.concurrent.Future from their send methods. This makes it challenging to write asynchronous non-blocking code given Future's limited interface. Since Kafka now requires Java 8, we now have the option of using CompletionStage and/or CompletableFuture that were introduced to solve this issue.

Also, this KIP is going to deprecate both send methods. The send method returning Future can be replaced by new API easily since CompletionStage is able to be converted to Future. The send method having callback can be replaced by CompletionStage#whenComplete.

Public Interfaces

Add the following method to org.apache.kafka.clients.producer.Producer. This is similar to send() but it returns java.util.concurrent.CompletionStage instead of java.util.concurrent.Future:

Producer

```
    CompletionStage<RecordMetadata> send(SendTarget target, SendRecord<K, V> record);
```

```
interface SendTarget {

  static TopicPartition of(String topic, int partition);

  static TopicTarget of(String topic);
}

class TopicTarget implements SendTarget

class TopicPartition implements SendTarget
```

```
interface SendRecord<K, V> {

  Headers headers();

  K key();

  V value();

  Long timestamp();

  static class Builder<K, V> {

    Builder<K, V> headers(Iterable<Header> value);
    Builder<K, V> key(String value);
    Builder<K, V> value(String value);
    Builder<K, V> timestamp(Long value);

    SendRecord<K, V> build();
  }
}
```

Proposed Changes

1. introduce new interface - SendTarget and SendRecord
2. introduce new method send(SendTarget, SendRecord) to producer

Compatibility, Deprecation, and Migration Plan

This change is binary compatible because `org.apache.kafka.clients.producer.Producer.send(ProducerRecord)` still returns `Future`. The use of covariant return types causes the bytecode to contain methods with the old and new signature.

Future work

We should encourage users to use new API and new interfaces (SendRecord and SendTarget). Hence, users should NOT create ProducerRecord anymore. We should deprecate all constructors of ProducerRecord.

Rejected Alternatives

- Return CompletableFuture from KafkaProducer.send
- Change send() to return CompletableFuture. This is not a compatible change.
- Return CompletionStage instead of CompletableFuture. The former doesn't extend the Future interface, so it's incompatible unless we introduce new methods.