

## Lab 5.6: Kafka Retry, in-flight, request timeout, and retry backoff

Welcome to the session 5 lab 6. The work for this lab is done in `~/kafka-training/lab5.6`. In this lab, you are going to set up Kafka Producer retries.

Please refer to the [Kafka course notes](#) for any updates or changes to this lab.

Find the latest version of this lab [here](#).

### Lab Adding Retries and Timeouts

In this lab we set up timeouts, set up retries, set up retry back off and change inflight messages to 1 so retries don't store records out of order.

### Change it and then run it

As before startup ZooKeeper if needed and three Kafka brokers. Then we will modify `StockPriceKafkaProducer` to configure retry, timeouts, in-flight message count and retry back off. We will run the `StockPriceKafkaProducer`. While we start and stop any two different Kafka Brokers while `StockPriceKafkaProducer` runs. Please notice retry messages in log of `StockPriceKafkaProducer`.

`~/kafka-training/lab5.6/src/main/java/com/cloudurable/kafka/producer/StockPriceKafkaProducer.java`

**Kafka Producer: StockPriceKafkaProducer disable batching**

```
public class StockPriceKafkaProducer {

    private static Producer<String, StockPrice>
        createProducer() {

        final Properties props = new Properties();
        setupBootstrapAndSerializers(props);
        setupBatchingAndCompression(props);
        setupRetriesInFlightTimeout(props);

        return new KafkaProducer<>(props);
    }

    ...

    private static void setupRetriesInFlightTimeout(Properties props) {
        //Only two in-flight messages per Kafka broker connection
        // - max.in.flight.requests.per.connection (default 5)
        props.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION,
            1);

        //Set the number of retries - retries
        props.put(ProducerConfig.RETRIES_CONFIG, 3);

        //Request timeout - request.timeout.ms
        props.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, 15_000);

        //Only retry after one second.
```

```
        props.put(ProducerConfig.RETRY_BACKOFF_MS_CONFIG, 1_000);  
    }  
    ...  
}
```

The above configures Producer retry, timeouts, in-flight message count and retry back off.

## Expected output after 2 broker shutdown

Run all brokers and then kill any two servers. Look for retry messages in the Producer log. Restart brokers and see the Producer recover. Also use replica verification to see when the broker catches up.

***ACTION*** - EDIT `StockPriceKafkaProducer` and finish the `setupRetriesInFlightTimeout` using instructions in the file.

***ACTION*** - START Kafka Brokers and ZooKeeper if needed.

***ACTION*** - Run `StockPriceKafkaProducer` from the IDE.

***ACTION*** - Run `SimpleStockPriceConsumer` from the IDE.

***ACTION*** - KILL one Kafka Broker while `StockPriceKafkaProducer` is running.

***ACTION*** - KILL two Kafka Broker while `StockPriceKafkaProducer` is running.

## WARN Inflight Message Count

The `MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION` is the max number of unacknowledged requests that a client can send on a single connection before blocking. If  $> 1$  and failed sends, then there is a risk of message re-ordering on partition during retry attempt (they could be written out of order of the send). If this is bad, depends on use but for our StockPrices this is not good, you should pick `retries > 1` or `inflight > 1` but not both. Avoid duplicates. The June 2017 release fixed this with sequence from producer.