package com.twitter.search.feature\_update\_service.modules;

import javax.inject.Named;

import javax.inject.Singleton;

import com.google.inject.Provides;

import com.twitter.app.Flaggable;

import com.twitter.common.util.Clock;

import com.twitter.finatra.kafka.producers.BlockingFinagleKafkaProducer;

import com.twitter.inject.TwitterModule;

import com.twitter.inject.annotations.Flag;

import com.twitter.search.common.indexing.thriftjava.ThriftVersionedEvents;

import com.twitter.search.common.util.io.kafka.CompactThriftSerializer;

import com.twitter.search.common.util.io.kafka.FinagleKafkaClientUtils;

import com.twitter.search.common.util.io.kafka.SearchPartitioner;

import com.twitter.search.common.util.io.kafka.SearchPartitionerRealtimeCg;

public class FinagleKafkaProducerModule extends TwitterModule {

public static final String KAFKA\_DEST\_FLAG = "kafka.dest";

public static final String KAFKA\_TOPIC\_NAME\_UPDATE\_EVENTS\_FLAG =

"kafka.topic.name.update\_events";

public static final String KAFKA\_TOPIC\_NAME\_UPDATE\_EVENTS\_FLAG\_REALTIME\_CG =

"kafka.topic.name.update\_events\_realtime\_cg";

public static final String KAFKA\_ENABLE\_S2S\_AUTH\_FLAG = "kafka.enable\_s2s\_auth";

public FinagleKafkaProducerModule() {

flag(KAFKA\_DEST\_FLAG, "Kafka cluster destination", "", Flaggable.ofString());

flag(KAFKA\_TOPIC\_NAME\_UPDATE\_EVENTS\_FLAG, "",

"Topic name for update events", Flaggable.ofString());

flag(KAFKA\_TOPIC\_NAME\_UPDATE\_EVENTS\_FLAG\_REALTIME\_CG, "",

"Topic name for update events", Flaggable.ofString());

flag(KAFKA\_ENABLE\_S2S\_AUTH\_FLAG, true, "enable s2s authentication configs",

Flaggable.ofBoolean());

}

@Provides

@Named("KafkaProducer")

public BlockingFinagleKafkaProducer<Long, ThriftVersionedEvents> kafkaProducer(

@Flag(KAFKA\_DEST\_FLAG) String kafkaDest,

@Flag(KAFKA\_ENABLE\_S2S\_AUTH\_FLAG) boolean enableKafkaAuth) {

return FinagleKafkaClientUtils.newFinagleKafkaProducer(

kafkaDest, enableKafkaAuth, new CompactThriftSerializer<ThriftVersionedEvents>(),

"search\_cluster", SearchPartitioner.class);

}

@Provides

@Named("KafkaProducerRealtimeCg")

public BlockingFinagleKafkaProducer<Long, ThriftVersionedEvents> kafkaProducerRealtimeCg(

@Flag(KAFKA\_DEST\_FLAG) String kafkaDest,

@Flag(KAFKA\_ENABLE\_S2S\_AUTH\_FLAG) boolean enableKafkaAuth) {

return FinagleKafkaClientUtils.newFinagleKafkaProducer(

kafkaDest, enableKafkaAuth, new CompactThriftSerializer<ThriftVersionedEvents>(),

"search\_cluster", SearchPartitionerRealtimeCg.class);

}

@Provides

@Singleton

public Clock clock() {

return Clock.SYSTEM\_CLOCK;

}

}