package com.twitter.search.earlybird.partition;

import com.google.common.annotations.VisibleForTesting;

import org.apache.kafka.clients.consumer.ConsumerRecord;

import org.apache.kafka.clients.consumer.KafkaConsumer;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.twitter.common.util.Clock;

import com.twitter.search.earlybird.exception.MissingKafkaTopicException;

import com.twitter.ubs.thriftjava.AudioSpaceBaseEvent;

import com.twitter.ubs.thriftjava.AudioSpaceEvent;

import com.twitter.util.Duration;

/\*\*

\*

\* An example publish event looks like this:

\* <AudioBaseSpaceEvent space\_publish\_event:SpacePublishEvent(

\* time\_stamp\_millis:1616430926899,

\* user\_id:123456,

\* broadcast\_id:123456789)>

\*/

public class AudioSpaceEventsStreamIndexer extends SimpleStreamIndexer<Long, AudioSpaceBaseEvent> {

private static final Logger LOG = LoggerFactory.getLogger(AudioSpaceEventsStreamIndexer.class);

private static final String AUDIO\_SPACE\_EVENTS\_TOPIC = "audio\_space\_events\_v1";

@VisibleForTesting

// We use this to filter out old space publish events so as to avoid the risk of processing

// old space publish events whose corresponding finish events are no longer in the stream.

// It's unlikely that spaces would last longer than this constant so it should be safe to assume

// that the space whose publish event is older than this age is finished.

protected static final long MAX\_PUBLISH\_EVENTS\_AGE\_MS =

Duration.fromHours(11).inMillis();

private final AudioSpaceTable audioSpaceTable;

private final Clock clock;

public AudioSpaceEventsStreamIndexer(

KafkaConsumer<Long, AudioSpaceBaseEvent> kafkaConsumer,

AudioSpaceTable audioSpaceTable,

Clock clock) throws MissingKafkaTopicException {

super(kafkaConsumer, AUDIO\_SPACE\_EVENTS\_TOPIC);

this.audioSpaceTable = audioSpaceTable;

this.clock = clock;

}

@Override

protected void validateAndIndexRecord(ConsumerRecord<Long, AudioSpaceBaseEvent> record) {

AudioSpaceBaseEvent baseEvent = record.value();

if (baseEvent != null && baseEvent.isSetBroadcast\_id() && baseEvent.isSetEvent\_metadata()) {

AudioSpaceEvent event = baseEvent.getEvent\_metadata();

String spaceId = baseEvent.getBroadcast\_id();

if (event != null && event.isSet(AudioSpaceEvent.\_Fields.SPACE\_PUBLISH\_EVENT)) {

long publishEventAgeMs = clock.nowMillis() - baseEvent.getTime\_stamp\_millis();

if (publishEventAgeMs < MAX\_PUBLISH\_EVENTS\_AGE\_MS) {

audioSpaceTable.audioSpaceStarts(spaceId);

}

} else if (event != null && event.isSet(AudioSpaceEvent.\_Fields.SPACE\_END\_EVENT)) {

audioSpaceTable.audioSpaceFinishes(spaceId);

}

}

}

@VisibleForTesting

public AudioSpaceTable getAudioSpaceTable() {

return audioSpaceTable;

}

void printSummary() {

LOG.info(audioSpaceTable.toString());

}

}