package com.twitter.search.ingester.pipeline.twitter.kafka;

import javax.naming.NamingException;

import com.google.common.base.Preconditions;

import org.apache.commons.pipeline.StageException;

import org.apache.commons.pipeline.validation.ConsumedTypes;

import com.twitter.search.ingester.model.IngesterTwitterMessage;

import com.twitter.search.ingester.pipeline.twitter.ThriftVersionedEventsConverter;

import com.twitter.search.ingester.pipeline.util.PipelineStageException;

@ConsumedTypes(IngesterTwitterMessage.class)

public class DeleteUpdateEventsKafkaProducerStage extends KafkaProducerStage

<IngesterTwitterMessage> {

private ThriftVersionedEventsConverter converter;

public DeleteUpdateEventsKafkaProducerStage() {

super();

}

public DeleteUpdateEventsKafkaProducerStage(String topicName, String clientId,

String clusterPath) {

super(topicName, clientId, clusterPath);

}

@Override

protected void innerSetup() throws PipelineStageException, NamingException {

super.innerSetup();

commonInnerSetup();

}

@Override

protected void doInnerPreprocess() throws StageException, NamingException {

super.doInnerPreprocess();

commonInnerSetup();

}

private void commonInnerSetup() throws NamingException {

converter = new ThriftVersionedEventsConverter(wireModule.getPenguinVersions());

}

@Override

public void innerProcess(Object obj) throws StageException {

if (!(obj instanceof IngesterTwitterMessage)) {

throw new StageException(this, "Object is not an IngesterTwitterMessage: " + obj);

}

IngesterTwitterMessage message = (IngesterTwitterMessage) obj;

innerRunFinalStageOfBranchV2(message);

}

@Override

protected void innerRunFinalStageOfBranchV2(IngesterTwitterMessage message) {

converter.updatePenguinVersions(wireModule.getCurrentlyEnabledPenguinVersions());

Preconditions.checkArgument(message.getFromUserTwitterId().isPresent(),

"Missing user ID.");

super.tryToSendEventsToKafka(converter.toDelete(

message.getTweetId(), message.getUserId(), message.getDebugEvents()));

}

}