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

import javax.naming.NamingException;

import org.apache.commons.pipeline.StageException;

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

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

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

import com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants;

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

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

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

@ConsumedTypes(IngesterTwitterMessage.class)

@ProducedTypes(ThriftVersionedEvents.class)

public class ConvertToThriftVersionedEventsStage extends TwitterBaseStage

<IngesterTwitterMessage, IngesterThriftVersionedEvents> {

private ThriftVersionedEventsConverter converter;

@Override

public void doInnerPreprocess() throws StageException, NamingException {

super.doInnerPreprocess();

innerSetup();

}

@Override

protected void innerSetup() 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 ingesterTwitterMessage = (IngesterTwitterMessage) obj;

IngesterThriftVersionedEvents maybeEvents = tryToConvert(ingesterTwitterMessage);

if (maybeEvents == null) {

throw new StageException(

this, "Object is not a retweet or a reply: " + ingesterTwitterMessage);

}

emitAndCount(maybeEvents);

}

@Override

protected IngesterThriftVersionedEvents innerRunStageV2(IngesterTwitterMessage message) {

IngesterThriftVersionedEvents maybeEvents = tryToConvert(message);

if (maybeEvents == null) {

throw new PipelineStageRuntimeException("Object is not a retweet or reply, does not have to"

+ " pass to next stage");

}

return maybeEvents;

}

private IngesterThriftVersionedEvents tryToConvert(IngesterTwitterMessage message) {

converter.updatePenguinVersions(wireModule.getCurrentlyEnabledPenguinVersions());

if (!message.isRetweet() && !message.isReplyToTweet()) {

return null;

}

if (message.isRetweet()) {

return converter.toOutOfOrderAppend(

message.getRetweetMessage().getSharedId(),

EarlybirdFieldConstants.EarlybirdFieldConstant.RETWEETED\_BY\_USER\_ID,

message.getUserId(),

message.getDebugEvents().deepCopy());

}

return converter.toOutOfOrderAppend(

message.getInReplyToStatusId().get(),

EarlybirdFieldConstants.EarlybirdFieldConstant.REPLIED\_TO\_BY\_USER\_ID,

message.getUserId(),

message.getDebugEvents().deepCopy());

}

}