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

import java.util.Collection;

import javax.naming.NamingException;

import org.apache.commons.pipeline.StageException;

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

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

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

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

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

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

import com.twitter.util.Future;

/\*\*

\* Read-only stage for looking up location info and populating it onto messages.

\*/

@ConsumedTypes(IngesterTwitterMessage.class)

@ProducesConsumed

public final class PopulateCodedLocationsBatchedStage

extends TwitterBatchedBaseStage<IngesterTwitterMessage, IngesterTwitterMessage> {

private static final String GEOCODE\_DATASET\_NAME = "ingester\_geocode\_profile\_location";

private ManhattanCodedLocationProvider manhattanCodedLocationProvider = null;

/\*\*

\* Require lat/lon from TwitterMessage instead of lookup from coded\_locations,

\* do not batch sql, and simply emit messages passed in with regions populated on them

\* rather than emitting to indexing queues.

\*/

@Override

protected void doInnerPreprocess() throws StageException, NamingException {

super.doInnerPreprocess();

commonInnerSetup();

}

@Override

protected void innerSetup() throws PipelineStageException, NamingException {

super.innerSetup();

commonInnerSetup();

}

private void commonInnerSetup() throws NamingException {

this.manhattanCodedLocationProvider = ManhattanCodedLocationProvider.createWithEndpoint(

wireModule.getJavaManhattanKVEndpoint(),

getStageNamePrefix(),

GEOCODE\_DATASET\_NAME);

}

@Override

public void initStats() {

super.initStats();

}

@Override

protected Class<IngesterTwitterMessage> getQueueObjectType() {

return IngesterTwitterMessage.class;

}

@Override

protected Future<Collection<IngesterTwitterMessage>> innerProcessBatch(Collection<BatchedElement

<IngesterTwitterMessage, IngesterTwitterMessage>> batch) {

Collection<IngesterTwitterMessage> batchedElements = extractOnlyElementsFromBatch(batch);

return manhattanCodedLocationProvider.populateCodedLatLon(batchedElements);

}

@Override

protected boolean needsToBeBatched(IngesterTwitterMessage message) {

return !message.hasGeoLocation() && (message.getLocation() != null)

&& !message.getLocation().isEmpty();

}

@Override

protected IngesterTwitterMessage transform(IngesterTwitterMessage element) {

return element;

}

}