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.PipelineStageException;

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

import com.twitter.util.Future;

@ConsumedTypes(IngesterTwitterMessage.class)

@ProducesConsumed

public class LookupUserPropertiesBatchedStage extends TwitterBatchedBaseStage

<IngesterTwitterMessage, IngesterTwitterMessage> {

protected UserPropertiesManager userPropertiesManager;

@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 userPropertiesManager.populateUserProperties(batchedElements);

}

@Override

protected boolean needsToBeBatched(IngesterTwitterMessage element) {

return true;

}

@Override

protected IngesterTwitterMessage transform(IngesterTwitterMessage element) {

return element;

}

@Override

public synchronized void doInnerPreprocess() throws StageException, NamingException {

super.doInnerPreprocess();

commonInnerSetup();

}

@Override

protected void innerSetup() throws PipelineStageException, NamingException {

super.innerSetup();

commonInnerSetup();

}

private void commonInnerSetup() throws NamingException {

userPropertiesManager = new UserPropertiesManager(wireModule.getMetastoreClient());

}

}