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

import java.util.function.Supplier;

import org.apache.commons.pipeline.Pipeline;

import org.apache.commons.pipeline.StageDriver;

import org.apache.commons.pipeline.StageException;

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

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

/\*\*

\* This stage is a shim for the UserUpdatesPipeline.

\*

\* Eventually the UserUpdatesPipeline will be called directly from a TwitterServer, but this exists

\* as a bridge while we migrate.

\*/

public class UserUpdatesPipelineStage extends TwitterBaseStage {

// This is 'prod', 'staging', or 'staging1'.

private String environment;

private UserUpdatesPipeline userUpdatesPipeline;

@Override

protected void doInnerPreprocess() throws StageException {

StageDriver driver = ((Pipeline) stageContext).getStageDriver(this);

Supplier<Boolean> booleanSupplier = () -> driver.getState() == StageDriver.State.RUNNING;

try {

userUpdatesPipeline = UserUpdatesPipeline.buildPipeline(

environment,

wireModule,

getStageNamePrefix(),

booleanSupplier,

clock);

} catch (Exception e) {

throw new StageException(this, e);

}

PipelineUtil.feedStartObjectToStage(this);

}

@Override

public void innerProcess(Object obj) throws StageException {

userUpdatesPipeline.run();

}

@SuppressWarnings("unused") // populated from pipeline config

public void setEnvironment(String environment) {

this.environment = environment;

}

}