package com.twitter.recos.hose.common

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.recos.internal.thriftscala.RecosHoseMessage

import com.twitter.util.Future

import org.apache.kafka.clients.consumer.ConsumerRecord

/\*\*

\* The class processes RecosHoseMessage and inserts the message as an edge into a recos graph.

\*/

case class RecosEdgeProcessor(

edgeCollector: EdgeCollector

)(

implicit statsReceiver: StatsReceiver) {

private val scopedStats = statsReceiver.scope("RecosEdgeProcessor")

private val processEventsCounter = scopedStats.counter("process\_events")

private val nullPointerEventCounter = scopedStats.counter("null\_pointer\_num")

private val errorCounter = scopedStats.counter("process\_errors")

def process(record: ConsumerRecord[String, RecosHoseMessage]): Future[Unit] = {

processEventsCounter.incr()

val message = record.value()

try {

// the message is nullable

if (message != null) {

edgeCollector.addEdge(message)

} else {

nullPointerEventCounter.incr()

}

Future.Unit

} catch {

case e: Throwable =>

errorCounter.incr()

e.printStackTrace()

Future.Unit

}

}

}