package com.twitter.unified\_user\_actions.service.module

import com.twitter.kafka.client.headers.ATLA

import com.twitter.kafka.client.headers.Implicits.\_

import com.twitter.kafka.client.headers.PDXA

import com.twitter.kafka.client.headers.Zone

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

object ZoneFiltering {

def zoneMapping(zone: String): Zone = zone.toLowerCase match {

case "atla" => ATLA

case "pdxa" => PDXA

case \_ =>

throw new IllegalArgumentException(

s"zone must be provided and must be one of [atla,pdxa], provided $zone")

}

def localDCFiltering[K, V](event: ConsumerRecord[K, V], localZone: Zone): Boolean =

event.headers().isLocalZone(localZone)

def noFiltering[K, V](event: ConsumerRecord[K, V], localZone: Zone): Boolean = true

}