package com.twitter.tweetypie

package handler

import com.twitter.eventbus.client.EventBusPublisher

import com.twitter.stitch.Stitch

import com.twitter.tweetypie.backends.GeoScrubEventStore.GetGeoScrubTimestamp

import com.twitter.tweetypie.thriftscala.DeleteLocationData

import com.twitter.tweetypie.thriftscala.DeleteLocationDataRequest

/\*\*

\* Initiates the process of removing the geo information from a user's

\* tweets.

\*/

object DeleteLocationDataHandler {

type Type = DeleteLocationDataRequest => Future[Unit]

def apply(

getLastScrubTime: GetGeoScrubTimestamp,

scribe: DeleteLocationData => Future[Unit],

eventbus: EventBusPublisher[DeleteLocationData]

): Type =

request => {

// Attempt to bound the time range of the tweets that need to be

// scrubbed by finding the most recent scrub time on record. This

// is an optimization that prevents scrubbing already-scrubbed

// tweets, so it is OK if the value that we find is occasionally

// stale or if the lookup fails. Primarily, this is intended to

// protect against intentional abuse by enqueueing multiple

// delete\_location\_data events that have to traverse a very long

// timeline.

Stitch

.run(getLastScrubTime(request.userId))

// If there is no timestamp or the lookup failed, continue with

// an unchanged request.

.handle { case \_ => None }

.flatMap { lastScrubTime =>

// Due to clock skew, it's possible for the last scrub

// timestamp to be larger than the timestamp from the request,

// but we ignore that so that we keep a faithful record of

// user requests. The execution of such events will end up a

// no-op.

val event =

DeleteLocationData(

userId = request.userId,

timestampMs = Time.now.inMilliseconds,

lastTimestampMs = lastScrubTime.map(\_.inMilliseconds)

)

Future.join(

Seq(

// Scribe the event so that we can reprocess events if

// there is a bug or operational issue that causes some

// events to be lost.

scribe(event),

// The actual deletion process is handled by the TweetyPie

// geoscrub daemon.

eventbus.publish(event)

)

)

}

}

}