package com.twitter.tweetypie

package hydrator

import com.twitter.tweetypie.core.\_

import com.twitter.tweetypie.repository.\_

import com.twitter.tweetypie.thriftscala.\_

/\*\*

\* This hydrator, which is really more of a "repairer", scrubs at read-time geo data

\* that should have been scrubbed but wasn't. For any tweet with geo data, it checks

\* the last geo-scrub timestamp, if any, for the user, and if the tweet was created before

\* that timestamp, it removes the geo data.

\*/

object GeoScrubHydrator {

type Data = (Option[GeoCoordinates], Option[PlaceId])

type Type = ValueHydrator[Data, TweetCtx]

private[this] val modifiedNoneNoneResult = ValueState.modified((None, None))

def apply(repo: GeoScrubTimestampRepository.Type, scribeTweetId: FutureEffect[TweetId]): Type =

ValueHydrator[Data, TweetCtx] { (curr, ctx) =>

repo(ctx.userId).liftToTry.map {

case Return(geoScrubTime) if ctx.createdAt <= geoScrubTime =>

scribeTweetId(ctx.tweetId)

modifiedNoneNoneResult

// no-op on failure and no result

case \_ => ValueState.unmodified(curr)

}

}.onlyIf { case ((coords, place), \_) => coords.nonEmpty || place.nonEmpty }

}