package com.twitter.tweetypie

package hydrator

import com.twitter.dataproducts.enrichments.thriftscala.ProfileGeoEnrichment

import com.twitter.tweetypie.core.\_

import com.twitter.tweetypie.repository.ProfileGeoKey

import com.twitter.tweetypie.repository.ProfileGeoRepository

import com.twitter.tweetypie.thriftscala.FieldByPath

object ProfileGeoHydrator {

type Type = ValueHydrator[Option[ProfileGeoEnrichment], TweetCtx]

val hydratedField: FieldByPath = fieldByPath(Tweet.ProfileGeoEnrichmentField)

private[this] val partialResult = ValueState.partial(None, hydratedField)

def apply(repo: ProfileGeoRepository.Type): Type =

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

val key =

ProfileGeoKey(

tweetId = ctx.tweetId,

userId = Some(ctx.userId),

coords = ctx.geoCoordinates

)

repo(key).liftToTry.map {

case Return(enrichment) => ValueState.modified(Some(enrichment))

case Throw(\_) => partialResult

}

}.onlyIf((curr, ctx) =>

curr.isEmpty && ctx.tweetFieldRequested(Tweet.ProfileGeoEnrichmentField))

}