package com.twitter.tweetypie.repository

import com.twitter.consumer\_privacy.mention\_controls.thriftscala.UnmentionInfo

import com.twitter.stitch.Stitch

import com.twitter.strato.client.Fetcher

import com.twitter.strato.client.{Client => StratoClient}

import com.twitter.tweetypie.thriftscala.Tweet

import com.twitter.strato.thrift.ScroogeConvImplicits.\_

object UnmentionInfoRepository {

type Type = Tweet => Stitch[Option[UnmentionInfo]]

val column = "consumer-privacy/mentions-management/unmentionInfoFromTweet"

case class UnmentionInfoView(asViewer: Option[Long])

/\*\*

\* Creates a function that extracts users fields from a tweet and checks

\* if the extracted users have been unmentioned from the tweet's asssociated conversation.

\* This function enables the prefetch caching of UnmentionInfo used by graphql during createTweet

\* events and mirrors the logic found in the unmentionInfo Strato column found

\* here: http://go/unmentionInfo.strato

\* @param client Strato client

\* @return

\*/

def apply(client: StratoClient): Type = {

val fetcher: Fetcher[Tweet, UnmentionInfoView, UnmentionInfo] =

client.fetcher[Tweet, UnmentionInfoView, UnmentionInfo](column)

tweet =>

tweet.coreData.flatMap(\_.conversationId) match {

case Some(conversationId) =>

val viewerUserId = TwitterContext().flatMap(\_.userId)

fetcher

.fetch(tweet, UnmentionInfoView(viewerUserId))

.map(\_.v)

case \_ => Stitch.None

}

}

}