package com.twitter.follow\_recommendations.common.clients.impression\_store

import com.twitter.follow\_recommendations.common.models.DisplayLocation

import com.twitter.follow\_recommendations.common.models.WtfImpression

import com.twitter.follow\_recommendations.thriftscala.{DisplayLocation => TDisplayLocation}

import com.twitter.stitch.Stitch

import com.twitter.strato.catalog.Scan.Slice

import com.twitter.strato.client.Scanner

import com.twitter.util.Time

import com.twitter.util.logging.Logging

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class WtfImpressionStore @Inject() (

scanner: Scanner[

((Long, TDisplayLocation), Slice[Long]),

Unit,

((Long, TDisplayLocation), Long),

(Long, Int)

]) extends Logging {

def get(userId: Long, dl: DisplayLocation): Stitch[Seq[WtfImpression]] = {

val thriftDl = dl.toThrift

scanner.scan(((userId, thriftDl), Slice.all[Long])).map { impressionsPerDl =>

val wtfImpressions =

for {

(((\_, \_), candidateId), (latestTs, counts)) <- impressionsPerDl

} yield WtfImpression(

candidateId = candidateId,

displayLocation = dl,

latestTime = Time.fromMilliseconds(latestTs),

counts = counts

)

wtfImpressions

} rescue {

// fail open so that the request can still go through

case ex: Throwable =>

logger.warn(s"$dl WtfImpressionsStore warn: " + ex.getMessage)

Stitch.Nil

}

}

}