package com.twitter.timelineranker.common

import com.twitter.search.common.constants.thriftscala.ThriftLanguage

import com.twitter.servo.util.FutureArrow

import com.twitter.timelineranker.model.RecapQuery

import com.twitter.timelines.clients.manhattan.LanguageUtils

import com.twitter.timelines.clients.manhattan.UserMetadataClient

import com.twitter.timelines.util.FailOpenHandler

import com.twitter.util.Future

import com.twitter.service.metastore.gen.thriftscala.UserLanguages

object UserLanguagesTransform {

val EmptyUserLanguagesFuture: Future[UserLanguages] =

Future.value(UserMetadataClient.EmptyUserLanguages)

}

/\*\*

\* FutureArrow which fetches user languages

\* It should be run in parallel with the main pipeline which fetches and hydrates CandidateTweets

\*/

class UserLanguagesTransform(handler: FailOpenHandler, userMetadataClient: UserMetadataClient)

extends FutureArrow[RecapQuery, Seq[ThriftLanguage]] {

override def apply(request: RecapQuery): Future[Seq[ThriftLanguage]] = {

import UserLanguagesTransform.\_

handler {

userMetadataClient.getUserLanguages(request.userId)

} { \_: Throwable => EmptyUserLanguagesFuture }

}.map(LanguageUtils.computeLanguages(\_))

}