package com.twitter.timelineranker.common

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.servo.util.FutureArrow

import com.twitter.timelineranker.core.CandidateEnvelope

import com.twitter.timelines.clients.relevance\_search.SearchClient

import com.twitter.timelines.model.TweetId

import com.twitter.util.Future

trait RecapHydrationSearchResultsTransformBase

extends FutureArrow[CandidateEnvelope, CandidateEnvelope] {

protected def statsReceiver: StatsReceiver

protected def searchClient: SearchClient

private[this] val numResultsFromSearchStat = statsReceiver.stat("numResultsFromSearch")

def tweetIdsToHydrate(envelope: CandidateEnvelope): Seq[TweetId]

override def apply(envelope: CandidateEnvelope): Future[CandidateEnvelope] = {

searchClient

.getTweetsScoredForRecap(

envelope.query.userId,

tweetIdsToHydrate(envelope),

envelope.query.earlybirdOptions

).map { results =>

numResultsFromSearchStat.add(results.size)

envelope.copy(searchResults = results)

}

}

}