package com.twitter.timelineranker.uteg\_liked\_by\_tweets

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.timelineranker.common.RecapHydrationSearchResultsTransformBase

import com.twitter.timelineranker.core.CandidateEnvelope

import com.twitter.timelineranker.model.RecapQuery.DependencyProvider

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

import com.twitter.timelines.model.TweetId

import com.twitter.util.Future

class UtegLikedByTweetsSearchResultsTransform(

override protected val searchClient: SearchClient,

override protected val statsReceiver: StatsReceiver,

relevanceSearchProvider: DependencyProvider[Boolean])

extends RecapHydrationSearchResultsTransformBase {

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

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

envelope.utegResults.keys.toSeq

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

searchClient

.getTweetsScoredForRecap(

userId = envelope.query.userId,

tweetIds = tweetIdsToHydrate(envelope),

earlybirdOptions = envelope.query.earlybirdOptions,

logSearchDebugInfo = false,

searchClientId = None,

relevanceSearch = relevanceSearchProvider(envelope.query)

).map { results =>

numResultsFromSearchStat.add(results.size)

envelope.copy(searchResults = results)

}

}

}