package com.twitter.search.earlybird\_root.mergers;

import java.util.List;

import java.util.concurrent.TimeUnit;

import com.google.common.base.Preconditions;

import com.twitter.search.common.metrics.SearchTimerStats;

import com.twitter.search.earlybird.thrift.EarlybirdResponse;

import com.twitter.search.earlybird.thrift.ThriftSearchQuery;

import com.twitter.search.earlybird.thrift.ThriftSearchRankingMode;

import com.twitter.search.earlybird.thrift.ThriftSearchResults;

import com.twitter.search.earlybird\_root.collectors.RelevanceMergeCollector;

import com.twitter.search.earlybird\_root.common.EarlybirdRequestContext;

import com.twitter.util.Future;

/\*\*

\* Merger class to merge toptweets EarlybirdResponse objects

\*/

public class TopTweetsResponseMerger extends EarlybirdResponseMerger {

private static final double SUCCESSFUL\_RESPONSE\_THRESHOLD = 0.9;

private static final SearchTimerStats TIMER =

SearchTimerStats.export("merge\_top\_tweets", TimeUnit.NANOSECONDS, false, true);

public TopTweetsResponseMerger(EarlybirdRequestContext requestContext,

List<Future<EarlybirdResponse>> responses,

ResponseAccumulator mode) {

super(requestContext, responses, mode);

}

@Override

protected SearchTimerStats getMergedResponseTimer() {

return TIMER;

}

@Override

protected double getDefaultSuccessResponseThreshold() {

return SUCCESSFUL\_RESPONSE\_THRESHOLD;

}

@Override

protected EarlybirdResponse internalMerge(EarlybirdResponse mergedResponse) {

final ThriftSearchQuery searchQuery = requestContext.getRequest().getSearchQuery();

Preconditions.checkNotNull(searchQuery);

Preconditions.checkState(searchQuery.isSetRankingMode());

Preconditions.checkState(searchQuery.getRankingMode() == ThriftSearchRankingMode.TOPTWEETS);

int numResultsRequested = computeNumResultsToKeep();

RelevanceMergeCollector collector = new RelevanceMergeCollector(responses.size());

addResponsesToCollector(collector);

ThriftSearchResults searchResults = collector.getAllSearchResults();

if (numResultsRequested < searchResults.getResults().size()) {

searchResults.setResults(searchResults.getResults().subList(0, numResultsRequested));

}

mergedResponse.setSearchResults(searchResults);

return mergedResponse;

}

}