package com.twitter.search.common.util.earlybird;

import java.util.Map;

import com.google.common.base.Function;

import com.google.common.collect.Iterables;

import com.google.common.collect.Maps;

/\*\*

\* Utility class used to help merging results.

\*/

public final class ResultsUtil {

private ResultsUtil() { }

/\*\*

\* Aggregate a list of responses in the following way.

\* 1. For each response, mapGetter can turn the response into a map.

\* 2. Dump all entries from the above map into a "total" map, which accumulates entries from

\* all the responses.

\*/

public static <T, V> Map<T, Integer> aggregateCountMap(

Iterable<V> responses,

Function<V, Map<T, Integer>> mapGetter) {

Map<T, Integer> total = Maps.newHashMap();

for (Map<T, Integer> map : Iterables.transform(responses, mapGetter)) {

if (map != null) {

for (Map.Entry<T, Integer> entry : map.entrySet()) {

T key = entry.getKey();

total.put(key, total.containsKey(key)

? total.get(key) + entry.getValue() : entry.getValue());

}

}

}

return total;

}

}