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

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

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

public final class PartitionResponseAccumulator extends ResponseAccumulator {

private static final String TARGET\_TYPE\_PARTITION = "partition";

@Override

public String getNameForLogging(int responseIndex, int numTotalResponses) {

return TARGET\_TYPE\_PARTITION + responseIndex;

}

@Override

public String getNameForEarlybirdResponseCodeStats(int responseIndex, int numTotalResponses) {

// We do not need to differentiate between partitions: we just want to get the number of

// responses returned by Earlybirds, for each EarlybirdResponseCode.

return TARGET\_TYPE\_PARTITION;

}

@Override

boolean shouldEarlyTerminateMerge(EarlyTerminateTierMergePredicate merger) {

return false;

}

@Override

public void handleSkippedResponse(EarlybirdResponseCode responseCode) { }

@Override

public void handleErrorResponse(EarlybirdResponse response) {

}

@Override

public AccumulatedResponses.PartitionCounts getPartitionCounts() {

return new AccumulatedResponses.PartitionCounts(getNumResponses(),

getSuccessResponses().size() + getSuccessfulEmptyResponseCount(), null);

}

@Override

protected boolean isMergingAcrossTiers() {

return false;

}

}