package com.twitter.search.core.earlybird.facets;

import java.util.Arrays;

import java.util.Map;

import com.google.common.collect.ImmutableMap;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.twitter.search.common.constants.thriftjava.ThriftLanguage;

/\*\*

\* A util class to build a language histogram

\*/

public class LanguageHistogram {

private static final Logger LOG = LoggerFactory.getLogger(LanguageHistogram.class);

public static final LanguageHistogram EMPTY\_HISTOGRAM = new LanguageHistogram() {

// Let's make this immutable for safety.

@Override public void clear() {

throw new UnsupportedOperationException();

}

@Override public void increment(int languageID) {

throw new UnsupportedOperationException();

}

@Override public void add(int languageID, int value) {

throw new UnsupportedOperationException();

}

@Override public void addAll(LanguageHistogram histogram) {

throw new UnsupportedOperationException();

}

};

private final int[] languageHistogram = new int[ThriftLanguage.values().length];

public int[] getLanguageHistogram() {

return languageHistogram;

}

/\*\*

\* Returns this histogram represented as a language->count map.

\*/

public Map<ThriftLanguage, Integer> getLanguageHistogramAsMap() {

ImmutableMap.Builder<ThriftLanguage, Integer> builder = ImmutableMap.builder();

for (int i = 0; i < languageHistogram.length; i++) {

// ThriftLanguage.findByValue() might return null, which should fall back to UNKNOWN.

ThriftLanguage lang = ThriftLanguage.findByValue(i);

lang = lang == null ? ThriftLanguage.UNKNOWN : lang;

builder.put(lang, languageHistogram[i]);

}

return builder.build();

}

public void clear() {

Arrays.fill(languageHistogram, 0);

}

public void increment(int languageId) {

if (isValidLanguageId(languageId)) {

languageHistogram[languageId]++;

}

}

public void increment(ThriftLanguage language) {

increment(language.getValue());

}

public void add(int languageId, int value) {

if (isValidLanguageId(languageId)) {

languageHistogram[languageId] += value;

}

}

public void add(ThriftLanguage language, int value) {

add(language.getValue(), value);

}

/\*\*

\* Adds all entries from the provided histogram to this histogram.

\*/

public void addAll(LanguageHistogram histogram) {

if (histogram == EMPTY\_HISTOGRAM) {

return;

}

for (int i = 0; i < languageHistogram.length; i++) {

languageHistogram[i] += histogram.languageHistogram[i];

}

}

// Check for out of bound languages. If a language is out of bounds, we don't want it

// to cause the entire search to fail.

private boolean isValidLanguageId(int languageId) {

if (languageId < languageHistogram.length) {

return true;

} else {

LOG.error("Language id " + languageId + " out of range");

return false;

}

}

}