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

import java.io.IOException;

import com.google.common.base.Preconditions;

import org.apache.lucene.index.NumericDocValues;

import com.twitter.search.common.schema.base.Schema;

import com.twitter.search.core.earlybird.index.EarlybirdIndexSegmentAtomicReader;

/\*\*

\* An iterator that looks up the termID from the appropriate CSF

\*/

public class CSFFacetCountIterator extends FacetCountIterator {

private final int fieldID;

private final NumericDocValues numericDocValues;

/\*\*

\* Creates a new iterator for the given facet csf field.

\*/

public CSFFacetCountIterator(

EarlybirdIndexSegmentAtomicReader reader,

Schema.FieldInfo facetFieldInfo) throws IOException {

FacetIDMap.FacetField facetField = reader.getFacetIDMap().getFacetField(facetFieldInfo);

Preconditions.checkNotNull(facetField);

this.fieldID = facetField.getFacetId();

numericDocValues = reader.getNumericDocValues(facetFieldInfo.getName());

Preconditions.checkNotNull(numericDocValues);

}

@Override

public void collect(int internalDocID) throws IOException {

if (numericDocValues.advanceExact(internalDocID)) {

long termID = numericDocValues.longValue();

if (shouldCollect(internalDocID, termID)) {

collect(internalDocID, termID, fieldID);

}

}

}

/\*\*

\* Subclasses should override if they need to restrict the docs or termIDs

\* that they collect on. For example, these may need to override if

\* 1) Not all docs set this field, so we should not collect on

\* the default value of 0

\* 2) The same CSF field means different things (in particular, shared\_status\_id means

\* retweet OR reply parent id) so we need to do some other check to determine if we should

\* collect

\*

\* @return whether we should collect on this doc/termID

\*/

protected boolean shouldCollect(int internalDocID, long termID) throws IOException {

return true;

}

}