package com.twitter.search.earlybird.index;

import java.io.IOException;

import org.apache.lucene.index.LeafReader;

import org.apache.lucene.index.Terms;

import org.apache.lucene.index.TermsEnum;

import org.apache.lucene.search.DocIdSetIterator;

import org.apache.lucene.util.BytesRef;

public final class DocValuesHelper {

private DocValuesHelper() {

}

/\*\*

\* Reverse lookup. Given a value, returns the first doc ID with this value. This requires a field

\* that indexes the values.

\*

\* @param reader The reader to use to look up field values.

\* @param value The value to lookup.

\* @param indexField The field containing an index of the values.

\*/

public static int getFirstDocIdWithValue(

LeafReader reader, String indexField, BytesRef value) throws IOException {

TermsEnum termsEnum = getTermsEnum(reader, indexField);

if (termsEnum == null || !termsEnum.seekExact(value)) {

return DocIdSetIterator.NO\_MORE\_DOCS;

}

DocIdSetIterator docsIterator = termsEnum.postings(null);

return docsIterator.nextDoc();

}

/\*\*

\* Reverse lookup. Same as getFirstDocIdWithValue(), but if no document with the given value

\* exists, the next bigger value is used for looking up the first doc ID.

\*

\* If there are multiple documents that match the value, all documents will be scanned, and the

\* largest doc ID that matches will be returned.

\*

\* @param reader The reader to use to look up field values.

\* @param value The value to lookup.

\* @param indexField The field containing an index of the values.

\*/

public static int getLargestDocIdWithCeilOfValue(

LeafReader reader, String indexField, BytesRef value) throws IOException {

TermsEnum termsEnum = getTermsEnum(reader, indexField);

if (termsEnum == null) {

return DocIdSetIterator.NO\_MORE\_DOCS;

}

if (termsEnum.seekCeil(value) == TermsEnum.SeekStatus.END) {

return DocIdSetIterator.NO\_MORE\_DOCS;

}

DocIdSetIterator docsIterator = termsEnum.postings(null);

int docId = docsIterator.nextDoc();

while (docsIterator.nextDoc() != DocIdSetIterator.NO\_MORE\_DOCS) {

docId = docsIterator.docID();

}

return docId;

}

private static TermsEnum getTermsEnum(LeafReader reader, String indexField) throws IOException {

Terms terms = reader.terms(indexField);

if (terms == null) {

return null;

}

return terms.iterator();

}

}