package com.twitter.search.common.query;

import java.io.IOException;

import com.google.common.base.Preconditions;

import org.apache.lucene.search.DocIdSetIterator;

import org.apache.lucene.search.Scorer;

import org.apache.lucene.search.Weight;

/\*\*

\* Scorer implementation that adds attribute collection support for an underlying query.

\* Meant to be used in conjunction with {@link IdentifiableQuery}.

\*/

public class IdentifiableQueryScorer extends FilteredScorer {

private final FieldRankHitInfo queryId;

private final HitAttributeCollector attrCollector;

public IdentifiableQueryScorer(Weight weight, Scorer inner, FieldRankHitInfo queryId,

HitAttributeCollector attrCollector) {

super(weight, inner);

this.queryId = queryId;

this.attrCollector = Preconditions.checkNotNull(attrCollector);

}

@Override

public DocIdSetIterator iterator() {

final DocIdSetIterator superDISI = super.iterator();

return new DocIdSetIterator() {

@Override

public int docID() {

return superDISI.docID();

}

@Override

public int nextDoc() throws IOException {

int docid = superDISI.nextDoc();

if (docid != NO\_MORE\_DOCS) {

attrCollector.collectScorerAttribution(docid, queryId);

}

return docid;

}

@Override

public int advance(int target) throws IOException {

int docid = superDISI.advance(target);

if (docid != NO\_MORE\_DOCS) {

attrCollector.collectScorerAttribution(docid, queryId);

}

return docid;

}

@Override

public long cost() {

return superDISI.cost();

}

};

}

}