package com.twitter.search.earlybird.search.relevance.scoring;

import org.apache.lucene.search.Explanation;

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

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

/\*

\* A sample scorer, doesn't really do anything, returns the same score for every document.

\*/

public class DefaultScoringFunction extends ScoringFunction {

private float score;

public DefaultScoringFunction(ImmutableSchemaInterface schema) {

super(schema);

}

@Override

protected float score(float luceneQueryScore) {

score = luceneQueryScore;

return luceneQueryScore;

}

@Override

protected Explanation doExplain(float luceneScore) {

// just an example - this scoring function will go away soon

return Explanation.match(luceneScore, "luceneScore=" + luceneScore);

}

@Override

public void updateRelevanceStats(ThriftSearchResultsRelevanceStats relevanceStats) {

relevanceStats.setNumScored(relevanceStats.getNumScored() + 1);

if (score == ScoringFunction.SKIP\_HIT) {

relevanceStats.setNumSkipped(relevanceStats.getNumSkipped() + 1);

}

}

}