package com.twitter.search.common.search.termination;

import java.io.IOException;

import java.util.Set;

import com.google.common.base.Preconditions;

import org.apache.lucene.index.LeafReaderContext;

import org.apache.lucene.index.Term;

import org.apache.lucene.search.Explanation;

import org.apache.lucene.search.Scorer;

import org.apache.lucene.search.Weight;

/\*\*

\* Weight implementation that adds termination support for an underlying query.

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

\*/

public class TerminationQueryWeight extends Weight {

private final Weight inner;

private final QueryTimeout timeout;

TerminationQueryWeight(TerminationQuery query, Weight inner, QueryTimeout timeout) {

super(query);

this.inner = inner;

this.timeout = Preconditions.checkNotNull(timeout);

}

@Override

public Explanation explain(LeafReaderContext context, int doc)

throws IOException {

return inner.explain(context, doc);

}

@Override

public Scorer scorer(LeafReaderContext context) throws IOException {

Scorer innerScorer = inner.scorer(context);

if (innerScorer != null) {

return new TerminationQueryScorer(this, innerScorer, timeout);

}

return null;

}

@Override

public void extractTerms(Set<Term> terms) {

inner.extractTerms(terms);

}

@Override

public boolean isCacheable(LeafReaderContext ctx) {

return inner.isCacheable(ctx);

}

}