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

import java.io.IOException;

import java.util.List;

import org.apache.lucene.facet.Facets;

import org.apache.lucene.facet.FacetsCollector;

import com.twitter.search.common.facets.CountFacetSearchParam;

import com.twitter.search.common.facets.FacetSearchParam;

import com.twitter.search.common.facets.FacetsFactory;

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

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

/\*\*

\* Factory for EarlybirdFacets

\*/

public class EarlybirdFacetsFactory implements FacetsFactory {

private final EarlybirdIndexSegmentAtomicReader reader;

public EarlybirdFacetsFactory(EarlybirdIndexSegmentAtomicReader reader) {

this.reader = reader;

}

@Override

public Facets create(

List<FacetSearchParam> facetSearchParams,

FacetsCollector facetsCollector) throws IOException {

return new EarlybirdFacets(facetSearchParams, facetsCollector, reader);

}

@Override

public boolean accept(FacetSearchParam facetSearchParam) {

if (!(facetSearchParam instanceof CountFacetSearchParam)

|| (facetSearchParam.getFacetFieldRequest().getPath() != null

&& !facetSearchParam.getFacetFieldRequest().getPath().isEmpty())) {

return false;

}

String field = facetSearchParam.getFacetFieldRequest().getField();

Schema.FieldInfo facetInfo = reader.getSegmentData().getSchema()

.getFacetFieldByFacetName(field);

return facetInfo != null

&& reader.getSegmentData().getPerFieldMap().containsKey(facetInfo.getName());

}

}