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

import java.util.ArrayList;

import java.util.List;

import com.google.common.collect.ImmutableList;

import com.google.common.collect.Sets;

import org.apache.commons.lang.StringUtils;

import com.twitter.search.common.schema.earlybird.EarlybirdFieldConstants.EarlybirdFieldConstant;

import com.twitter.search.earlybird.partition.AudioSpaceTable;

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

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

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

public class SpaceFacetCollector extends AbstractFacetTermCollector {

private final List<ThriftSearchResultAudioSpace> spaces = new ArrayList<>();

private final AudioSpaceTable audioSpaceTable;

public SpaceFacetCollector(AudioSpaceTable audioSpaceTable) {

this.audioSpaceTable = audioSpaceTable;

}

@Override

public boolean collect(int docID, long termID, int fieldID) {

String spaceId = getTermFromFacet(termID, fieldID,

Sets.newHashSet(EarlybirdFieldConstant.SPACES\_FACET));

if (StringUtils.isEmpty(spaceId)) {

return false;

}

spaces.add(new ThriftSearchResultAudioSpace(spaceId,

audioSpaceTable.isRunning(spaceId) ? AudioSpaceState.RUNNING

: AudioSpaceState.ENDED));

return true;

}

@Override

public void fillResultAndClear(ThriftSearchResult result) {

getExtraMetadata(result).setSpaces(ImmutableList.copyOf(spaces));

spaces.clear();

}

}