package com.twitter.search.common.search;

import java.util.LinkedHashSet;

import java.util.Set;

import org.apache.lucene.search.Query;

import org.apache.lucene.spatial.prefix.tree.Cell;

import org.apache.lucene.spatial.prefix.tree.CellIterator;

import org.apache.lucene.util.BytesRef;

import com.twitter.search.common.util.spatial.GeohashChunkImpl;

import com.twitter.search.queryparser.util.GeoCode;

import geo.google.datamodel.GeoAddressAccuracy;

public final class GeoQuadTreeQueryBuilderUtil {

private GeoQuadTreeQueryBuilderUtil() {

}

/\*\*

\* Build a geo quad tree query based around the geo code based on the geo field.

\* @param geocode the geo location for the quad tree query

\* @param field the field where the geohash tokens are indexed

\* @return the corresponding for the geo quad tree query

\*/

public static Query buildGeoQuadTreeQuery(GeoCode geocode, String field) {

Set<BytesRef> geoHashSet = new LinkedHashSet<>();

// if accuracy is specified. Add a term query based on accuracy.

if (geocode.accuracy != GeoAddressAccuracy.UNKNOWN\_LOCATION.getCode()) {

BytesRef termRef = new BytesRef(GeohashChunkImpl.buildGeoStringWithAccuracy(geocode.latitude,

geocode.longitude,

geocode.accuracy));

geoHashSet.add(termRef);

}

// If distance is specified. Add term queries based on distance

if (geocode.distanceKm != GeoCode.DOUBLE\_DISTANCE\_NOT\_SET) {

// Build query based on distance

int treeLevel = -1;

// First find block containing query point with diagonal greater than 2 \* radius.

Cell centerNode = GeohashChunkImpl.getGeoNodeByRadius(geocode.latitude, geocode.longitude,

geocode.distanceKm);

// Add center node querying term

if (centerNode != null) {

geoHashSet.add(centerNode.getTokenBytesNoLeaf(new BytesRef()));

treeLevel = centerNode.getLevel();

}

// This improves edge case recall, by adding cells also intersecting the query area.

CellIterator nodes = GeohashChunkImpl.getNodesIntersectingCircle(geocode.latitude,

geocode.longitude,

geocode.distanceKm,

treeLevel);

// If there are other nodes intersecting query circle, also add them in.

if (nodes != null) {

while (nodes.hasNext()) {

geoHashSet.add(nodes.next().getTokenBytesNoLeaf(new BytesRef()));

}

}

}

return new com.twitter.search.common.query.MultiTermDisjunctionQuery(field, geoHashSet);

}

}