package com.twitter.follow\_recommendations.common.candidate\_sources.geo

import com.google.inject.Singleton

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.hermit.model.Algorithm

import com.twitter.product\_mixer.core.model.common.identifier.CandidateSourceIdentifier

import javax.inject.Inject

@Singleton

class PopGeohashSource @Inject() (

popGeoSource: PopGeoSource,

statsReceiver: StatsReceiver)

extends BasePopGeohashSource(

popGeoSource = popGeoSource,

statsReceiver = statsReceiver.scope("PopGeohashSource"),

) {

override def candidateSourceEnabled(target: Target): Boolean = true

override val identifier: CandidateSourceIdentifier = PopGeohashSource.Identifier

override def minGeohashLength(target: Target): Int = {

target.params(PopGeoSourceParams.PopGeoSourceGeoHashMinPrecision)

}

override def maxResults(target: Target): Int = {

target.params(PopGeoSourceParams.PopGeoSourceMaxResultsPerPrecision)

}

override def maxGeohashLength(target: Target): Int = {

target.params(PopGeoSourceParams.PopGeoSourceGeoHashMaxPrecision)

}

override def returnResultFromAllPrecision(target: Target): Boolean = {

target.params(PopGeoSourceParams.PopGeoSourceReturnFromAllPrecisions)

}

}

object PopGeohashSource {

val Identifier: CandidateSourceIdentifier = CandidateSourceIdentifier(

Algorithm.PopGeohash.toString)

}