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

import com.google.inject.Singleton

import com.google.inject.name.Named

import com.twitter.conversions.DurationOps.\_

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.follow\_recommendations.common.candidate\_sources.base.CachedCandidateSource

import com.twitter.follow\_recommendations.common.candidate\_sources.base.StratoFetcherWithUnitViewSource

import com.twitter.follow\_recommendations.common.constants.GuiceNamedConstants

import com.twitter.follow\_recommendations.common.models.AccountProof

import com.twitter.follow\_recommendations.common.models.CandidateUser

import com.twitter.follow\_recommendations.common.models.PopularInGeoProof

import com.twitter.follow\_recommendations.common.models.Reason

import com.twitter.hermit.pop\_geo.thriftscala.PopUsersInPlace

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

import com.twitter.strato.client.Fetcher

import com.twitter.util.Duration

import javax.inject.Inject

@Singleton

class BasePopGeoSource @Inject() (

@Named(GuiceNamedConstants.POP\_USERS\_IN\_PLACE\_FETCHER) fetcher: Fetcher[

String,

Unit,

PopUsersInPlace

]) extends StratoFetcherWithUnitViewSource[String, PopUsersInPlace](

fetcher,

BasePopGeoSource.Identifier) {

override def map(target: String, candidates: PopUsersInPlace): Seq[CandidateUser] =

BasePopGeoSource.map(target, candidates)

}

object BasePopGeoSource {

val Identifier: CandidateSourceIdentifier = CandidateSourceIdentifier("BasePopGeoSource")

val MaxResults = 200

def map(target: String, candidates: PopUsersInPlace): Seq[CandidateUser] =

candidates.popUsers.sortBy(-\_.score).take(BasePopGeoSource.MaxResults).view.map { candidate =>

CandidateUser(

id = candidate.userId,

score = Some(candidate.score),

reason = Some(

Reason(

Some(

AccountProof(

popularInGeoProof = Some(PopularInGeoProof(location = candidates.place))

)

)

)

)

)

}

}

@Singleton

class PopGeoSource @Inject() (basePopGeoSource: BasePopGeoSource, statsReceiver: StatsReceiver)

extends CachedCandidateSource[String, CandidateUser](

basePopGeoSource,

PopGeoSource.MaxCacheSize,

PopGeoSource.CacheTTL,

statsReceiver,

PopGeoSource.Identifier)

object PopGeoSource {

val Identifier: CandidateSourceIdentifier = CandidateSourceIdentifier("PopGeoSource")

val MaxCacheSize = 20000

val CacheTTL: Duration = 1.hours

}