package com.twitter.follow\_recommendations.utils

import com.twitter.follow\_recommendations.common.candidate\_sources.addressbook.\_

import com.twitter.follow\_recommendations.common.candidate\_sources.geo.PopCountrySource

import com.twitter.follow\_recommendations.common.candidate\_sources.geo.PopCountryBackFillSource

import com.twitter.follow\_recommendations.common.candidate\_sources.geo.PopGeoSource

import com.twitter.follow\_recommendations.common.candidate\_sources.geo.PopGeohashSource

import com.twitter.follow\_recommendations.common.candidate\_sources.ppmi\_locale\_follow.PPMILocaleFollowSource

import com.twitter.follow\_recommendations.common.candidate\_sources.recent\_engagement.RecentEngagementNonDirectFollowSource

import com.twitter.follow\_recommendations.common.candidate\_sources.sims.SwitchingSimsSource

import com.twitter.follow\_recommendations.common.candidate\_sources.sims\_expansion.RecentEngagementSimilarUsersSource

import com.twitter.follow\_recommendations.common.candidate\_sources.sims\_expansion.RecentFollowingSimilarUsersSource

import com.twitter.follow\_recommendations.common.candidate\_sources.sims\_expansion.RecentStrongEngagementDirectFollowSimilarUsersSource

import com.twitter.follow\_recommendations.common.candidate\_sources.socialgraph.RecentFollowingRecentFollowingExpansionSource

import com.twitter.follow\_recommendations.common.candidate\_sources.stp.MutualFollowStrongTiePredictionSource

import com.twitter.follow\_recommendations.common.candidate\_sources.stp.OfflineStrongTiePredictionSource

import com.twitter.follow\_recommendations.common.candidate\_sources.stp.BaseOnlineSTPSource

import com.twitter.follow\_recommendations.common.candidate\_sources.stp.SocialProofEnforcedOfflineStrongTiePredictionSource

import com.twitter.follow\_recommendations.common.candidate\_sources.triangular\_loops.TriangularLoopsSource

import com.twitter.follow\_recommendations.common.candidate\_sources.two\_hop\_random\_walk.TwoHopRandomWalkSource

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

import com.twitter.follow\_recommendations.configapi.params.GlobalParams

import com.twitter.follow\_recommendations.models.CandidateSourceType

import com.twitter.product\_mixer.core.functional\_component.candidate\_source.CandidateSource

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

import com.twitter.timelines.configapi.HasParams

trait CandidateSourceHoldbackUtil {

import CandidateSourceHoldbackUtil.\_

def filterCandidateSources[T <: HasParams](

request: T,

sources: Seq[CandidateSource[T, CandidateUser]]

): Seq[CandidateSource[T, CandidateUser]] = {

val typeToFilter = request.params(GlobalParams.CandidateSourcesToFilter)

val sourcesToFilter = CandidateSourceTypeToMap.get(typeToFilter).getOrElse(Set.empty)

sources.filterNot { source => sourcesToFilter.contains(source.identifier) }

}

}

object CandidateSourceHoldbackUtil {

final val ContextualActivityCandidateSourceIds: Set[CandidateSourceIdentifier] =

Set(

RecentFollowingSimilarUsersSource.Identifier,

RecentEngagementNonDirectFollowSource.Identifier,

RecentEngagementSimilarUsersSource.Identifier,

RecentStrongEngagementDirectFollowSimilarUsersSource.Identifier,

SwitchingSimsSource.Identifier,

)

final val SocialCandidateSourceIds: Set[CandidateSourceIdentifier] =

Set(

ForwardEmailBookSource.Identifier,

ForwardPhoneBookSource.Identifier,

ReverseEmailBookSource.Identifier,

ReversePhoneBookSource.Identifier,

RecentFollowingRecentFollowingExpansionSource.Identifier,

BaseOnlineSTPSource.Identifier,

MutualFollowStrongTiePredictionSource.Identifier,

OfflineStrongTiePredictionSource.Identifier,

SocialProofEnforcedOfflineStrongTiePredictionSource.Identifier,

TriangularLoopsSource.Identifier,

TwoHopRandomWalkSource.Identifier

)

final val GeoCandidateSourceIds: Set[CandidateSourceIdentifier] =

Set(

PPMILocaleFollowSource.Identifier,

PopCountrySource.Identifier,

PopGeohashSource.Identifier,

PopCountryBackFillSource.Identifier,

PopGeoSource.Identifier,

)

final val CandidateSourceTypeToMap: Map[CandidateSourceType.Value, Set[

CandidateSourceIdentifier

]] =

Map(

CandidateSourceType.Social -> SocialCandidateSourceIds,

CandidateSourceType.ActivityContextual -> ContextualActivityCandidateSourceIds,

CandidateSourceType.GeoAndInterests -> GeoCandidateSourceIds

)

}