package com.twitter.follow\_recommendations.common.predicates.sgs

import com.twitter.follow\_recommendations.common.base.Predicate

import com.twitter.follow\_recommendations.common.base.PredicateResult

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

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

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

import com.twitter.stitch.Stitch

import javax.inject.Singleton

@Singleton

class RecentFollowingPredicate extends Predicate[(HasRecentFollowedUserIds, CandidateUser)] {

override def apply(pair: (HasRecentFollowedUserIds, CandidateUser)): Stitch[PredicateResult] = {

val (targetUser, candidate) = pair

targetUser.recentFollowedUserIdsSet match {

case Some(users) =>

if (!users.contains(candidate.id)) {

RecentFollowingPredicate.ValidStitch

} else {

RecentFollowingPredicate.AlreadyFollowedStitch

}

case None => RecentFollowingPredicate.ValidStitch

}

}

}

object RecentFollowingPredicate {

val ValidStitch: Stitch[PredicateResult.Valid.type] = Stitch.value(PredicateResult.Valid)

val AlreadyFollowedStitch: Stitch[PredicateResult.Invalid] =

Stitch.value(PredicateResult.Invalid(Set(FilterReason.AlreadyFollowed)))

}