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

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.HasPreviousRecommendationsContext

import com.twitter.stitch.Stitch

import javax.inject.Singleton

@Singleton

class PreviouslyRecommendedUserIdsPredicate

extends Predicate[(HasPreviousRecommendationsContext, CandidateUser)] {

override def apply(

pair: (HasPreviousRecommendationsContext, CandidateUser)

): Stitch[PredicateResult] = {

val (targetUser, candidate) = pair

val previouslyRecommendedUserIDs = targetUser.previouslyRecommendedUserIDs

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

PreviouslyRecommendedUserIdsPredicate.ValidStitch

} else {

PreviouslyRecommendedUserIdsPredicate.AlreadyRecommendedStitch

}

}

}

object PreviouslyRecommendedUserIdsPredicate {

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

val AlreadyRecommendedStitch: Stitch[PredicateResult.Invalid] =

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

}