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.HasInvalidRelationshipUserIds

import com.twitter.stitch.Stitch

import javax.inject.Singleton

@Singleton

class InvalidRelationshipPredicate

extends Predicate[(HasInvalidRelationshipUserIds, CandidateUser)] {

override def apply(

pair: (HasInvalidRelationshipUserIds, CandidateUser)

): Stitch[PredicateResult] = {

val (targetUser, candidate) = pair

targetUser.invalidRelationshipUserIds match {

case Some(users) =>

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

InvalidRelationshipPredicate.ValidStitch

} else {

Stitch.value(InvalidRelationshipPredicate.InvalidRelationshipStitch)

}

case None => Stitch.value(PredicateResult.Valid)

}

}

}

object InvalidRelationshipPredicate {

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

val InvalidRelationshipStitch: PredicateResult.Invalid =

PredicateResult.Invalid(Set(FilterReason.InvalidRelationship))

}