package com.twitter.frigate.pushservice.refresh\_handler.cross

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.SocialContextActions

import com.twitter.frigate.pushservice.model.PushTypes.RawCandidate

import com.twitter.hermit.predicate.Predicate

class CopyPredicates(statsReceiver: StatsReceiver) {

val alwaysTruePredicate = Predicate

.from { \_: CandidateCopyPair =>

true

}.withStats(statsReceiver.scope("always\_true\_copy\_predicate"))

val unrecognizedCandidatePredicate = alwaysTruePredicate.flip

.withStats(statsReceiver.scope("unrecognized\_candidate"))

val displaySocialContextPredicate = Predicate

.from { candidateCopyPair: CandidateCopyPair =>

candidateCopyPair.candidate match {

case candidateWithScActions: RawCandidate with SocialContextActions =>

val socialContextUserIds = candidateWithScActions.socialContextActions.map(\_.userId)

val countSocialContext = socialContextUserIds.size

val pushCopy = candidateCopyPair.pushCopy

countSocialContext match {

case 1 => pushCopy.hasOneDisplaySocialContext && !pushCopy.hasOtherSocialContext

case 2 => pushCopy.hasTwoDisplayContext && !pushCopy.hasOtherSocialContext

case c if c > 2 =>

pushCopy.hasOneDisplaySocialContext && pushCopy.hasOtherSocialContext

case \_ => false

}

case \_ => false

}

}.withStats(statsReceiver.scope("display\_social\_context\_predicate"))

}