package com.twitter.frigate.pushservice.predicate.ntab\_caret\_fatigue

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.predicate.ntab\_caret\_fatigue.NtabCaretClickFatiguePredicateHelper

import com.twitter.frigate.common.rec\_types.RecTypes

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

import com.twitter.hermit.predicate.NamedPredicate

object MagicFanoutNtabCaretFatiguePredicate {

val name = "MagicFanoutNtabCaretFatiguePredicateForCandidate"

private val MomentsCategory = "Moments"

private val MomentsViaMagicRecsCategory = "MomentsViaMagicRecs"

def apply()(implicit globalStats: StatsReceiver): NamedPredicate[PushCandidate] = {

val scopedStats = globalStats.scope(name)

val genericTypeCategories = Seq(MomentsCategory, MomentsViaMagicRecsCategory)

val crts = RecTypes.magicFanoutEventTypes

RecTypeNtabCaretClickFatiguePredicate

.apply(

genericTypeCategories,

crts,

NtabCaretClickFatiguePredicateHelper.calculateFatiguePeriodMagicRecs,

useMostRecentDislikeTime = true,

name = name

).withStats(scopedStats).withName(name)

}

}