package com.twitter.frigate.pushservice.predicate

import com.twitter.finagle.stats.StatsReceiver

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

import com.twitter.frigate.pushservice.params.PushFeatureSwitchParams.QualityPredicateIdParam

import com.twitter.frigate.pushservice.predicate.quality\_model\_predicate.\_

import com.twitter.hermit.predicate.NamedPredicate

import com.twitter.hermit.predicate.Predicate

import com.twitter.util.Future

object JointDauAndQualityModelPredicate {

val name = "JointDauAndQualityModelPredicate"

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

val stats = statsReceiver.scope(s"predicate\_$name")

val defaultPred = WeightedOpenOrNtabClickQualityPredicate()

val qualityPredicateMap = QualityPredicateMap()

Predicate

.fromAsync { candidate: PushCandidate =>

if (!candidate.target.skipModelPredicate) {

val modelPredicate =

qualityPredicateMap.getOrElse(

candidate.target.params(QualityPredicateIdParam),

defaultPred)

val modelPredicateResultFut =

modelPredicate.apply(Seq(candidate)).map(\_.headOption.getOrElse(false))

modelPredicateResultFut

} else Future.True

}

.withStats(stats)

.withName(name)

}

}