package com.twitter.frigate.pushservice.model

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.MagicFanoutProductLaunchCandidate

import com.twitter.frigate.common.util.{FeatureSwitchParams => FS}

import com.twitter.frigate.magic\_events.thriftscala.MagicEventsReason

import com.twitter.frigate.magic\_events.thriftscala.ProductType

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

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

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

import com.twitter.frigate.pushservice.predicate.magic\_fanout.MagicFanoutPredicatesUtil

import com.twitter.frigate.pushservice.config.Config

import com.twitter.frigate.pushservice.ml.PushMLModelScorer

import com.twitter.frigate.pushservice.model.candidate.CopyIds

import com.twitter.frigate.pushservice.model.ibis.MagicFanoutProductLaunchIbis2Hydrator

import com.twitter.frigate.pushservice.model.ntab.MagicFanoutProductLaunchNtabRequestHydrator

import com.twitter.frigate.pushservice.predicate.PredicatesForCandidate

import com.twitter.frigate.pushservice.predicate.magic\_fanout.MagicFanoutPredicatesForCandidate

import com.twitter.frigate.pushservice.predicate.ntab\_caret\_fatigue.MagicFanoutNtabCaretFatiguePredicate

import com.twitter.frigate.pushservice.take.predicates.BasicSendHandlerPredicates

import com.twitter.frigate.thriftscala.FrigateNotification

import com.twitter.hermit.predicate.NamedPredicate

class MagicFanoutProductLaunchPushCandidate(

candidate: RawCandidate with MagicFanoutProductLaunchCandidate,

copyIds: CopyIds

)(

implicit val statsScoped: StatsReceiver,

pushModelScorer: PushMLModelScorer)

extends PushCandidate

with MagicFanoutProductLaunchCandidate

with MagicFanoutProductLaunchIbis2Hydrator

with MagicFanoutProductLaunchNtabRequestHydrator {

override val frigateNotification: FrigateNotification = candidate.frigateNotification

override val pushCopyId: Option[Int] = copyIds.pushCopyId

override val ntabCopyId: Option[Int] = copyIds.ntabCopyId

override val pushId: Long = candidate.pushId

override val productLaunchType: ProductType = candidate.productLaunchType

override val candidateMagicEventsReasons: Seq[MagicEventsReason] =

candidate.candidateMagicEventsReasons

override val copyAggregationId: Option[String] = copyIds.aggregationId

override val target: Target = candidate.target

override val weightedOpenOrNtabClickModelScorer: PushMLModelScorer = pushModelScorer

override val statsReceiver: StatsReceiver =

statsScoped.scope("MagicFanoutProductLaunchPushCandidate")

}

case class MagicFanoutProductLaunchPushCandidatePredicates(config: Config)

extends BasicSendHandlerPredicates[MagicFanoutProductLaunchPushCandidate] {

implicit val statsReceiver: StatsReceiver = config.statsReceiver.scope(getClass.getSimpleName)

override val preCandidateSpecificPredicates: List[

NamedPredicate[MagicFanoutProductLaunchPushCandidate]

] =

List(

PredicatesForCandidate.isDeviceEligibleForCreatorPush,

PredicatesForCandidate.exceptedPredicate(

"excepted\_is\_target\_blue\_verified",

MagicFanoutPredicatesUtil.shouldSkipBlueVerifiedCheckForCandidate,

PredicatesForCandidate.isTargetBlueVerified.flip

), // no need to send if target is already Blue Verified

PredicatesForCandidate.exceptedPredicate(

"excepted\_is\_target\_legacy\_verified",

MagicFanoutPredicatesUtil.shouldSkipLegacyVerifiedCheckForCandidate,

PredicatesForCandidate.isTargetLegacyVerified.flip

), // no need to send if target is already Legacy Verified

PredicatesForCandidate.exceptedPredicate(

"excepted\_is\_target\_super\_follow\_creator",

MagicFanoutPredicatesUtil.shouldSkipSuperFollowCreatorCheckForCandidate,

PredicatesForCandidate.isTargetSuperFollowCreator.flip

), // no need to send if target is already Super Follow Creator

PredicatesForCandidate.paramPredicate(

FS.EnableMagicFanoutProductLaunch

),

MagicFanoutPredicatesForCandidate.magicFanoutProductLaunchFatigue(),

)

override val postCandidateSpecificPredicates: List[

NamedPredicate[MagicFanoutProductLaunchPushCandidate]

] =

List(

MagicFanoutNtabCaretFatiguePredicate(),

)

}