package com.twitter.frigate.pushservice.model

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.DiscoverTwitterCandidate

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.config.Config

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

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

import com.twitter.frigate.pushservice.model.ibis.DiscoverTwitterPushIbis2Hydrator

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

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

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

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

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

import com.twitter.frigate.thriftscala.CommonRecommendationType

import com.twitter.hermit.predicate.NamedPredicate

class DiscoverTwitterPushCandidate(

candidate: RawCandidate with DiscoverTwitterCandidate,

copyIds: CopyIds,

)(

implicit val statsScoped: StatsReceiver,

pushModelScorer: PushMLModelScorer)

extends PushCandidate

with DiscoverTwitterCandidate

with DiscoverTwitterPushIbis2Hydrator

with DiscoverTwitterNtabRequestHydrator {

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

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

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

override val target: Target = candidate.target

override lazy val commonRecType: CommonRecommendationType = candidate.commonRecType

override val weightedOpenOrNtabClickModelScorer: PushMLModelScorer = pushModelScorer

override val statsReceiver: StatsReceiver =

statsScoped.scope("DiscoverTwitterPushCandidate")

}

case class AddressBookPushCandidatePredicates(config: Config)

extends BasicRFPHPredicates[DiscoverTwitterPushCandidate] {

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

override val predicates: List[

NamedPredicate[DiscoverTwitterPushCandidate]

] =

List(

PredicatesForCandidate.paramPredicate(

PushFeatureSwitchParams.EnableAddressBookPush

)

)

}

case class CompleteOnboardingPushCandidatePredicates(config: Config)

extends BasicRFPHPredicates[DiscoverTwitterPushCandidate] {

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

override val predicates: List[

NamedPredicate[DiscoverTwitterPushCandidate]

] =

List(

PredicatesForCandidate.paramPredicate(

PushFeatureSwitchParams.EnableCompleteOnboardingPush

)

)

}

case class PopGeoTweetCandidatePredicates(override val config: Config)

extends OutOfNetworkTweetPredicates[OutOfNetworkTweetPushCandidate] {

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

override def postCandidateSpecificPredicates: List[

NamedPredicate[OutOfNetworkTweetPushCandidate]

] = List(

PredicatesForCandidate.htlFatiguePredicate(

PushFeatureSwitchParams.NewUserPlaybookAllowedLastLoginHours

)

)

}