package com.twitter.frigate.pushservice.model

import com.twitter.channels.common.thriftscala.ApiList

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.ListPushCandidate

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

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

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.ListIbis2Hydrator

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

import com.twitter.frigate.pushservice.predicate.ListPredicates

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

import com.twitter.frigate.thriftscala.CommonRecommendationType

import com.twitter.hermit.predicate.NamedPredicate

import com.twitter.storehaus.ReadableStore

import com.twitter.util.Future

class ListRecommendationPushCandidate(

val apiListStore: ReadableStore[Long, ApiList],

candidate: RawCandidate with ListPushCandidate,

copyIds: CopyIds

)(

implicit stats: StatsReceiver,

pushModelScorer: PushMLModelScorer)

extends PushCandidate

with ListPushCandidate

with ListIbis2Hydrator

with ListCandidateNTabRequestHydrator {

override val commonRecType: CommonRecommendationType = candidate.commonRecType

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

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

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

override val statsReceiver: StatsReceiver = stats

override val weightedOpenOrNtabClickModelScorer: PushMLModelScorer = pushModelScorer

override val target: PushTypes.Target = candidate.target

override val listId: Long = candidate.listId

lazy val apiList: Future[Option[ApiList]] = apiListStore.get(listId)

lazy val listName: Future[Option[String]] = apiList.map { apiListOpt =>

apiListOpt.map(\_.name)

}

lazy val listOwnerId: Future[Option[Long]] = apiList.map { apiListOpt =>

apiListOpt.map(\_.ownerId)

}

}

case class ListRecommendationPredicates(config: Config)

extends BasicRFPHPredicates[ListRecommendationPushCandidate] {

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

override val predicates: List[NamedPredicate[ListRecommendationPushCandidate]] = List(

ListPredicates.listNameExistsPredicate(),

ListPredicates.listAuthorExistsPredicate(),

ListPredicates.listAuthorAcceptableToTargetUser(config.edgeStore),

ListPredicates.listAcceptablePredicate(),

ListPredicates.listSubscriberCountPredicate()

)

}