package com.twitter.frigate.pushservice.model

import com.twitter.contentrecommender.thriftscala.MetricTag

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.OutOfNetworkTweetCandidate

import com.twitter.frigate.common.base.TopicCandidate

import com.twitter.frigate.common.base.TweetAuthorDetails

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.OutOfNetworkTweetIbis2HydratorForCandidate

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

import com.twitter.frigate.pushservice.predicate.HealthPredicates

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

import com.twitter.frigate.thriftscala.CommonRecommendationType

import com.twitter.gizmoduck.thriftscala.User

import com.twitter.hermit.predicate.NamedPredicate

import com.twitter.stitch.tweetypie.TweetyPie

import com.twitter.topiclisting.utt.LocalizedEntity

import com.twitter.util.Future

class OutOfNetworkTweetPushCandidate(

candidate: RawCandidate with OutOfNetworkTweetCandidate with TopicCandidate,

author: Future[Option[User]],

copyIds: CopyIds

)(

implicit stats: StatsReceiver,

pushModelScorer: PushMLModelScorer)

extends PushCandidate

with OutOfNetworkTweetCandidate

with TopicCandidate

with TweetAuthorDetails

with OutOfNetworkTweetNTabRequestHydrator

with OutOfNetworkTweetIbis2HydratorForCandidate {

override val statsReceiver: StatsReceiver = stats

override val weightedOpenOrNtabClickModelScorer: PushMLModelScorer = pushModelScorer

override val tweetId: Long = candidate.tweetId

override lazy val tweetyPieResult: Option[TweetyPie.TweetyPieResult] =

candidate.tweetyPieResult

override lazy val tweetAuthor: Future[Option[User]] = author

override val target: PushTypes.Target = candidate.target

override lazy 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 lazy val semanticCoreEntityId: Option[Long] = candidate.semanticCoreEntityId

override lazy val localizedUttEntity: Option[LocalizedEntity] = candidate.localizedUttEntity

override lazy val algorithmCR: Option[String] = candidate.algorithmCR

override lazy val isMrBackfillCR: Option[Boolean] = candidate.isMrBackfillCR

override lazy val tagsCR: Option[Seq[MetricTag]] = candidate.tagsCR

}

case class OutOfNetworkTweetCandidatePredicates(override val config: Config)

extends OutOfNetworkTweetPredicates[OutOfNetworkTweetPushCandidate] {

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

override def postCandidateSpecificPredicates: List[

NamedPredicate[OutOfNetworkTweetPushCandidate]

] =

List(

HealthPredicates.agathaAbusiveTweetAuthorPredicateMrTwistly(),

)

}