package com.twitter.frigate.pushservice.model

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.TripCandidate

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.take.predicates.OutOfNetworkTweetPredicates

import com.twitter.frigate.thriftscala.CommonRecommendationType

import com.twitter.gizmoduck.thriftscala.User

import com.twitter.stitch.tweetypie.TweetyPie

import com.twitter.topiclisting.utt.LocalizedEntity

import com.twitter.trends.trip\_v1.trip\_tweets.thriftscala.TripDomain

import com.twitter.util.Future

class TripTweetPushCandidate(

candidate: RawCandidate with OutOfNetworkTweetCandidate with TripCandidate,

author: Future[Option[User]],

copyIds: CopyIds

)(

implicit stats: StatsReceiver,

pushModelScorer: PushMLModelScorer)

extends PushCandidate

with TripCandidate

with TopicCandidate

with OutOfNetworkTweetCandidate

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] = None

override lazy val localizedUttEntity: Option[LocalizedEntity] = None

override lazy val algorithmCR: Option[String] = None

override val tripDomain: Option[collection.Set[TripDomain]] = candidate.tripDomain

}

case class TripTweetCandidatePredicates(override val config: Config)

extends OutOfNetworkTweetPredicates[TripTweetPushCandidate] {

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

}