package com.twitter.frigate.pushservice.model

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.SubscribedSearchTweetCandidate

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.SubscribedSearchTweetIbis2Hydrator

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

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

import com.twitter.gizmoduck.thriftscala.User

import com.twitter.stitch.tweetypie.TweetyPie

import com.twitter.util.Future

class SubscribedSearchTweetPushCandidate(

candidate: RawCandidate with SubscribedSearchTweetCandidate,

author: Option[User],

copyIds: CopyIds

)(

implicit stats: StatsReceiver,

pushModelScorer: PushMLModelScorer)

extends PushCandidate

with SubscribedSearchTweetCandidate

with TweetAuthorDetails

with SubscribedSearchTweetIbis2Hydrator

with SubscribedSearchTweetNtabRequestHydrator {

override def tweetAuthor: Future[Option[User]] = Future.value(author)

override def weightedOpenOrNtabClickModelScorer: PushMLModelScorer = pushModelScorer

override def tweetId: Long = candidate.tweetId

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

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

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

override def target: PushTypes.Target = candidate.target

override def searchTerm: String = candidate.searchTerm

override def timeBoundedLandingUrl: Option[String] = None

override def statsReceiver: StatsReceiver = stats

override def tweetyPieResult: Option[TweetyPie.TweetyPieResult] = candidate.tweetyPieResult

}

case class SubscribedSearchTweetCandidatePredicates(override val config: Config)

extends BasicTweetPredicatesForRFPH[SubscribedSearchTweetPushCandidate] {

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

}