package com.twitter.frigate.pushservice.model

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.SocialContextAction

import com.twitter.frigate.common.base.SocialContextUserDetails

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

import com.twitter.frigate.common.base.TweetRetweetCandidate

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

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

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

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

import com.twitter.frigate.pushservice.model.ibis.TweetRetweetCandidateIbis2Hydrator

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

import com.twitter.frigate.pushservice.util.CandidateHydrationUtil.TweetWithSocialContextTraits

import com.twitter.frigate.thriftscala.CommonRecommendationType

import com.twitter.gizmoduck.thriftscala.User

import com.twitter.stitch.tweetypie.TweetyPie

import com.twitter.util.Future

class TweetRetweetPushCandidate(

candidate: RawCandidate with TweetWithSocialContextTraits,

socialContextUserMap: Future[Map[Long, Option[User]]],

author: Future[Option[User]],

copyIds: CopyIds

)(

implicit stats: StatsReceiver,

pushModelScorer: PushMLModelScorer)

extends PushCandidate

with TweetRetweetCandidate

with SocialContextUserDetails

with TweetAuthorDetails

with TweetRetweetNTabRequestHydrator

with TweetRetweetCandidateIbis2Hydrator {

override val statsReceiver: StatsReceiver = stats

override val weightedOpenOrNtabClickModelScorer: PushMLModelScorer = pushModelScorer

override val tweetId: Long = candidate.tweetId

override val socialContextActions: Seq[SocialContextAction] =

candidate.socialContextActions

override val socialContextAllTypeActions: Seq[SocialContextAction] =

candidate.socialContextAllTypeActions

override lazy val scUserMap: Future[Map[Long, Option[User]]] = socialContextUserMap

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

override lazy val commonRecType: CommonRecommendationType = candidate.commonRecType

override val target: PushTypes.Target = candidate.target

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

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

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

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

}