package com.twitter.frigate.pushservice.model

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.F1FirstDegree

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

import com.twitter.frigate.common.base.SocialGraphServiceRelationshipMap

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

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

import com.twitter.frigate.pushservice.model.PushTypes.\_

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.F1FirstDegreeTweetIbis2HydratorForCandidate

import com.twitter.frigate.pushservice.model.ntab.F1FirstDegreeTweetNTabRequestHydrator

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

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

import com.twitter.frigate.thriftscala.CommonRecommendationType

import com.twitter.gizmoduck.thriftscala.User

import com.twitter.hermit.predicate.socialgraph.RelationEdge

import com.twitter.stitch.tweetypie.TweetyPie

import com.twitter.util.Future

class F1TweetPushCandidate(

candidate: RawCandidate with TweetWithSocialContextTraits,

author: Future[Option[User]],

socialGraphServiceResultMap: Map[RelationEdge, Boolean],

copyIds: CopyIds

)(

implicit stats: StatsReceiver,

pushModelScorer: PushMLModelScorer)

extends PushCandidate

with F1FirstDegree

with TweetAuthorDetails

with SocialGraphServiceRelationshipMap

with F1FirstDegreeTweetNTabRequestHydrator

with F1FirstDegreeTweetIbis2HydratorForCandidate {

override val socialContextActions: Seq[SocialContextAction] =

candidate.socialContextActions

override val socialContextAllTypeActions: Seq[SocialContextAction] =

candidate.socialContextActions

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 val relationshipMap: Map[RelationEdge, Boolean] = socialGraphServiceResultMap

}

case class F1TweetCandidatePredicates(override val config: Config)

extends BasicTweetPredicatesForRFPHWithoutSGSPredicates[F1TweetPushCandidate] {

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

}