package com.twitter.frigate.pushservice.model.ntab

import com.twitter.frigate.common.base.SocialContextActions

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

import com.twitter.frigate.common.base.TweetAuthor

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

import com.twitter.frigate.common.base.TweetCandidate

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

import com.twitter.frigate.pushservice.take.NotificationServiceSender

import com.twitter.notificationservice.thriftscala.DisplayTextEntity

import com.twitter.util.Future

trait TweetRetweetNTabRequestHydrator extends TweetNTabRequestHydrator with NTabSocialContext {

self: PushCandidate

with TweetCandidate

with TweetAuthor

with TweetAuthorDetails

with SocialContextActions

with SocialContextUserDetails =>

override lazy val displayTextEntitiesFut: Future[Seq[DisplayTextEntity]] = {

Future

.join(

NotificationServiceSender

.getDisplayTextEntityFromUser(tweetAuthor, "tweetAuthorName", isBold = false),

NotificationServiceSender

.generateSocialContextTextEntities(

rankedNtabDisplayNamesAndIds(defaultToRecency = false),

otherCount)

)

.map {

case (authorDisplay, socialContextDisplay) =>

socialContextDisplay ++ authorDisplay

}

}

override lazy val facepileUsersFut: Future[Seq[Long]] = Future.value(socialContextUserIds)

}