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

import com.twitter.frigate.pushservice.model.SubscribedSearchTweetPushCandidate

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

import com.twitter.notificationservice.thriftscala.DisplayText

import com.twitter.notificationservice.thriftscala.DisplayTextEntity

import com.twitter.notificationservice.thriftscala.TextValue

import com.twitter.util.Future

trait SubscribedSearchTweetNtabRequestHydrator extends TweetNTabRequestHydrator {

self: SubscribedSearchTweetPushCandidate =>

override def displayTextEntitiesFut: Future[Seq[DisplayTextEntity]] = NotificationServiceSender

.getDisplayTextEntityFromUser(tweetAuthor, "tweetAuthor", isBold = true).map(\_.toSeq)

override def socialProofDisplayText: Option[DisplayText] = {

Some(DisplayText(values = Seq(DisplayTextEntity("search\_query", TextValue.Text(searchTerm)))))

}

override lazy val facepileUsersFut: Future[Seq[Long]] = senderIdFut.map(Seq(\_))

override lazy val tapThroughFut: Future[String] =

Future.value(self.ntabLandingUrl)

}