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

import com.twitter.frigate.pushservice.model.ListRecommendationPushCandidate

import com.twitter.notificationservice.thriftscala.DisplayText

import com.twitter.notificationservice.thriftscala.DisplayTextEntity

import com.twitter.notificationservice.thriftscala.InlineCard

import com.twitter.notificationservice.thriftscala.StoryContext

import com.twitter.notificationservice.thriftscala.TextValue

import com.twitter.util.Future

trait ListCandidateNTabRequestHydrator extends NTabRequestHydrator {

self: ListRecommendationPushCandidate =>

override lazy val senderIdFut: Future[Long] =

listOwnerId.map(\_.getOrElse(0L))

override lazy val facepileUsersFut: Future[Seq[Long]] = Future.Nil

override lazy val storyContext: Option[StoryContext] = None

override lazy val inlineCard: Option[InlineCard] = None

override lazy val tapThroughFut: Future[String] = Future.value(s"i/lists/${listId}")

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

listNameOpt =>

listNameOpt.toSeq.map { name =>

DisplayTextEntity(name = "title", value = TextValue.Text(name))

}

}

override val socialProofDisplayText: Option[DisplayText] = Some(DisplayText())

}