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

import com.twitter.frigate.common.base.TopTweetImpressionsCandidate

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

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.StoryContextValue

import com.twitter.notificationservice.thriftscala.TextValue

import com.twitter.util.Future

trait TopTweetImpressionsNTabRequestHydrator extends NTabRequestHydrator {

self: PushCandidate with TopTweetImpressionsCandidate =>

override lazy val tapThroughFut: Future[String] =

Future.value(s"${target.targetId}/status/$tweetId")

override val senderIdFut: Future[Long] = Future.value(0L)

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

override val storyContext: Option[StoryContext] =

Some(StoryContext(altText = "", value = Some(StoryContextValue.Tweets(Seq(tweetId)))))

override val inlineCard: Option[InlineCard] = None

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

Future.value(

Seq(

DisplayTextEntity(name = "num\_impressions", value = TextValue.Number(self.impressionsCount))

)

)

}

override def socialProofDisplayText: Option[DisplayText] = None

}