package com.twitter.frigate.pushservice.model.ibis

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

import com.twitter.frigate.common.base.TweetRetweetCandidate

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

import com.twitter.frigate.pushservice.params.PushFeatureSwitchParams

import com.twitter.frigate.pushservice.util.PushIbisUtil.mergeModelValues

import com.twitter.util.Future

trait TweetRetweetCandidateIbis2Hydrator

extends TweetCandidateIbis2Hydrator

with RankedSocialContextIbis2Hydrator {

self: PushCandidate with TweetRetweetCandidate with TweetAuthorDetails =>

override lazy val tweetModelValues: Future[Map[String, String]] =

for {

socialContextModelValues <- socialContextModelValues

superModelValues <- super.tweetModelValues

tweetInlineModelValues <- tweetInlineActionModelValue

} yield {

superModelValues ++ mediaModelValue ++ otherModelValues ++ socialContextModelValues ++ tweetInlineModelValues ++ inlineVideoMediaMap

}

lazy val socialContextForRetweetMap: Map[String, String] =

if (self.target.params(PushFeatureSwitchParams.EnableSocialContextForRetweet)) {

Map("enable\_social\_context\_retweet" -> "true")

} else Map.empty[String, String]

override lazy val customFieldsMapFut: Future[Map[String, String]] =

mergeModelValues(super.customFieldsMapFut, socialContextForRetweetMap)

}