package com.twitter.frigate.pushservice.predicate

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.TweetDetails

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

import com.twitter.hermit.predicate.tweetypie.UserLocationAndTweet

import com.twitter.hermit.predicate.tweetypie.WithheldTweetPredicate

import com.twitter.hermit.predicate.NamedPredicate

import com.twitter.hermit.predicate.Predicate

import com.twitter.service.metastore.gen.thriftscala.Location

import com.twitter.util.Future

object TweetWithheldContentPredicate {

val name = "withheld\_content"

val defaultLocation = Location(city = "", region = "", countryCode = "", confidence = 0.0)

def apply(

)(

implicit statsReceiver: StatsReceiver

): NamedPredicate[PushCandidate with TweetDetails] = {

Predicate

.fromAsync { candidate: PushCandidate with TweetDetails =>

candidate.tweet match {

case Some(tweet) =>

WithheldTweetPredicate(checkAllCountries = true)

.apply(Seq(UserLocationAndTweet(defaultLocation, tweet)))

.map(\_.head)

case None =>

Future.value(false)

}

}

.withStats(statsReceiver.scope(s"predicate\_$name"))

.withName(name)

}

}