package com.twitter.visibility.builder.users

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.gizmoduck.thriftscala.User

import com.twitter.stitch.Stitch

import com.twitter.visibility.builder.FeatureMapBuilder

import com.twitter.visibility.common.UserDeviceSource

import com.twitter.visibility.features.AuthorHasConfirmedEmail

import com.twitter.visibility.features.AuthorHasVerifiedPhone

class AuthorDeviceFeatures(userDeviceSource: UserDeviceSource, statsReceiver: StatsReceiver) {

private[this] val scopedStatsReceiver = statsReceiver.scope("author\_device\_features")

private[this] val requests = scopedStatsReceiver.counter("requests")

private[this] val authorHasConfirmedEmailRequests =

scopedStatsReceiver.scope(AuthorHasConfirmedEmail.name).counter("requests")

private[this] val authorHasVerifiedPhoneRequests =

scopedStatsReceiver.scope(AuthorHasVerifiedPhone.name).counter("requests")

def forAuthor(author: User): FeatureMapBuilder => FeatureMapBuilder = forAuthorId(author.id)

def forAuthorId(authorId: Long): FeatureMapBuilder => FeatureMapBuilder = {

requests.incr()

\_.withFeature(AuthorHasConfirmedEmail, authorHasConfirmedEmail(authorId))

.withFeature(AuthorHasVerifiedPhone, authorHasVerifiedPhone(authorId))

}

def authorHasConfirmedEmail(authorId: Long): Stitch[Boolean] = {

authorHasConfirmedEmailRequests.incr()

userDeviceSource.hasConfirmedEmail(authorId)

}

def authorHasVerifiedPhone(authorId: Long): Stitch[Boolean] = {

authorHasVerifiedPhoneRequests.incr()

userDeviceSource.hasConfirmedPhone(authorId)

}

}