package com.twitter.visibility.builder.users

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.stitch.Stitch

import com.twitter.stitch.NotFound

import com.twitter.visibility.builder.FeatureMapBuilder

import com.twitter.visibility.common.UserId

import com.twitter.visibility.common.UserSensitiveMediaSettingsSource

import com.twitter.visibility.features.ViewerId

import com.twitter.visibility.features.ViewerSensitiveMediaSettings

import com.twitter.visibility.models.UserSensitiveMediaSettings

class ViewerSensitiveMediaSettingsFeatures(

userSensitiveMediaSettingsSource: UserSensitiveMediaSettingsSource,

statsReceiver: StatsReceiver) {

private[this] val scopedStatsReceiver =

statsReceiver.scope("viewer\_sensitive\_media\_settings\_features")

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

def forViewerId(viewerId: Option[UserId]): FeatureMapBuilder => FeatureMapBuilder = { builder =>

requests.incr()

builder

.withConstantFeature(ViewerId, viewerId)

.withFeature(ViewerSensitiveMediaSettings, viewerSensitiveMediaSettings(viewerId))

}

def viewerSensitiveMediaSettings(viewerId: Option[UserId]): Stitch[UserSensitiveMediaSettings] = {

(viewerId match {

case Some(userId) =>

userSensitiveMediaSettingsSource

.userSensitiveMediaSettings(userId)

.handle {

case NotFound => None

}

case \_ => Stitch.value(None)

}).map(UserSensitiveMediaSettings)

}

}