package com.twitter.follow\_recommendations.controllers

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.follow\_recommendations.common.models.CandidateUser

import com.twitter.follow\_recommendations.common.models.ClientContextConverter

import com.twitter.follow\_recommendations.common.models.DebugOptions

import com.twitter.follow\_recommendations.common.models.DisplayLocation

import com.twitter.follow\_recommendations.models.DebugParams

import com.twitter.follow\_recommendations.models.ScoringUserRequest

import com.twitter.timelines.configapi.Params

import javax.inject.Inject

import javax.inject.Singleton

import com.twitter.follow\_recommendations.{thriftscala => t}

import com.twitter.gizmoduck.thriftscala.UserType

import com.twitter.stitch.Stitch

@Singleton

class ScoringUserRequestBuilder @Inject() (

requestBuilderUserFetcher: RequestBuilderUserFetcher,

candidateUserDebugParamsBuilder: CandidateUserDebugParamsBuilder,

statsReceiver: StatsReceiver) {

private val scopedStats = statsReceiver.scope(this.getClass.getSimpleName)

private val isSoftUserCounter = scopedStats.counter("is\_soft\_user")

def fromThrift(req: t.ScoringUserRequest): Stitch[ScoringUserRequest] = {

requestBuilderUserFetcher.fetchUser(req.clientContext.userId).map { userOpt =>

val isSoftUser = userOpt.exists(\_.userType == UserType.Soft)

if (isSoftUser) isSoftUserCounter.incr()

val candidateUsersParamsMap = candidateUserDebugParamsBuilder.fromThrift(req)

val candidates = req.candidates.map { candidate =>

CandidateUser

.fromUserRecommendation(candidate).copy(params =

candidateUsersParamsMap.paramsMap.getOrElse(candidate.userId, Params.Invalid))

}

ScoringUserRequest(

clientContext = ClientContextConverter.fromThrift(req.clientContext),

displayLocation = DisplayLocation.fromThrift(req.displayLocation),

params = Params.Empty,

debugOptions = req.debugParams.map(DebugOptions.fromDebugParamsThrift),

recentFollowedUserIds = None,

recentFollowedByUserIds = None,

wtfImpressions = None,

similarToUserIds = Nil,

candidates = candidates,

debugParams = req.debugParams.map(DebugParams.fromThrift),

isSoftUser = isSoftUser

)

}

}

}