package com.twitter.follow\_recommendations.controllers

import com.twitter.decider.Decider

import com.twitter.decider.SimpleRecipient

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.follow\_recommendations.common.base.StatsUtil

import com.twitter.follow\_recommendations.configapi.deciders.DeciderKey

import com.twitter.gizmoduck.thriftscala.LookupContext

import com.twitter.gizmoduck.thriftscala.User

import com.twitter.stitch.Stitch

import com.twitter.stitch.gizmoduck.Gizmoduck

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class RequestBuilderUserFetcher @Inject() (

gizmoduck: Gizmoduck,

statsReceiver: StatsReceiver,

decider: Decider) {

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

def fetchUser(userIdOpt: Option[Long]): Stitch[Option[User]] = {

userIdOpt match {

case Some(userId) if enableDecider(userId) =>

val stitch = gizmoduck

.getUserById(

userId = userId,

context = LookupContext(

forUserId = Some(userId),

includeProtected = true,

includeSoftUsers = true

)

).map(user => Some(user))

StatsUtil

.profileStitch(stitch, scopedStats)

.handle {

case \_: Throwable => None

}

case \_ => Stitch.None

}

}

private def enableDecider(userId: Long): Boolean = {

decider.isAvailable(

DeciderKey.EnableFetchUserInRequestBuilder.toString,

Some(SimpleRecipient(userId)))

}

}