package com.twitter.follow\_recommendations.common.clients.phone\_storage\_service

import com.twitter.cds.contact\_consent\_state.thriftscala.PurposeOfProcessing

import com.twitter.phonestorage.api.thriftscala.GetUserPhonesByUsersRequest

import com.twitter.phonestorage.api.thriftscala.PhoneStorageService

import com.twitter.stitch.Stitch

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class PhoneStorageServiceClient @Inject() (

val phoneStorageService: PhoneStorageService.MethodPerEndpoint) {

/\*\*

\* PSS can potentially return multiple phone records.

\* The current implementation of getUserPhonesByUsers returns only a single phone for a single user\_id but

\* we can trivially support handling multiple in case that changes in the future.

\*/

def getPhoneNumbers(

userId: Long,

purposeOfProcessing: PurposeOfProcessing,

forceCarrierLookup: Option[Boolean] = None

): Stitch[Seq[String]] = {

val req = GetUserPhonesByUsersRequest(

userIds = Seq(userId),

forceCarrierLookup = forceCarrierLookup,

purposesOfProcessing = Some(Seq(purposeOfProcessing))

)

Stitch.callFuture(phoneStorageService.getUserPhonesByUsers(req)) map {

\_.userPhones.map(\_.phoneNumber)

}

}

}