package com.twitter.follow\_recommendations.common.models

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

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

import com.twitter.product\_mixer.core.model.marshalling.request.ClientContext

object ClientContextConverter {

def toFRSOfflineClientContextThrift(

productMixerClientContext: ClientContext

): offline.OfflineClientContext =

offline.OfflineClientContext(

productMixerClientContext.userId,

productMixerClientContext.guestId,

productMixerClientContext.appId,

productMixerClientContext.countryCode,

productMixerClientContext.languageCode,

productMixerClientContext.guestIdAds,

productMixerClientContext.guestIdMarketing

)

def fromThrift(clientContext: frs.ClientContext): ClientContext = ClientContext(

userId = clientContext.userId,

guestId = clientContext.guestId,

appId = clientContext.appId,

ipAddress = clientContext.ipAddress,

userAgent = clientContext.userAgent,

countryCode = clientContext.countryCode,

languageCode = clientContext.languageCode,

isTwoffice = clientContext.isTwoffice,

userRoles = clientContext.userRoles.map(\_.toSet),

deviceId = clientContext.deviceId,

guestIdAds = clientContext.guestIdAds,

guestIdMarketing = clientContext.guestIdMarketing,

mobileDeviceId = None,

mobileDeviceAdId = None,

limitAdTracking = None

)

def toThrift(clientContext: ClientContext): frs.ClientContext = frs.ClientContext(

userId = clientContext.userId,

guestId = clientContext.guestIdAds,

appId = clientContext.appId,

ipAddress = clientContext.ipAddress,

userAgent = clientContext.userAgent,

countryCode = clientContext.countryCode,

languageCode = clientContext.languageCode,

isTwoffice = clientContext.isTwoffice,

userRoles = clientContext.userRoles,

deviceId = clientContext.deviceId,

guestIdAds = clientContext.guestIdAds,

guestIdMarketing = clientContext.guestIdMarketing

)

}