package com.twitter.product\_mixer.core.model.marshalling.request

import com.fasterxml.jackson.annotation.JsonIgnore

import com.twitter.product\_mixer.core.pipeline.pipeline\_failure.BadRequest

import com.twitter.product\_mixer.core.pipeline.pipeline\_failure.PipelineFailure

/\*\*

\* ClientContext contains fields related to the client making the request.

\*/

case class ClientContext(

userId: Option[Long],

guestId: Option[Long],

guestIdAds: Option[Long],

guestIdMarketing: Option[Long],

appId: Option[Long],

ipAddress: Option[String],

userAgent: Option[String],

countryCode: Option[String],

languageCode: Option[String],

isTwoffice: Option[Boolean],

userRoles: Option[Set[String]],

deviceId: Option[String],

mobileDeviceId: Option[String],

mobileDeviceAdId: Option[String],

limitAdTracking: Option[Boolean])

object ClientContext {

val empty: ClientContext = ClientContext(

userId = None,

guestId = None,

guestIdAds = None,

guestIdMarketing = None,

appId = None,

ipAddress = None,

userAgent = None,

countryCode = None,

languageCode = None,

isTwoffice = None,

userRoles = None,

deviceId = None,

mobileDeviceId = None,

mobileDeviceAdId = None,

limitAdTracking = None

)

}

/\*\*

\* HasClientContext indicates that a request has [[ClientContext]] and adds helper functions for

\* accessing [[ClientContext]] fields.

\*/

trait HasClientContext {

def clientContext: ClientContext

/\*\*

\* getRequiredUserId returns a userId and throw if it's missing.

\*

\* @note logged out requests are disabled by default so this is safe for most products

\*/

@JsonIgnore /\*\* Jackson tries to serialize this method, throwing an exception for guest products \*/

def getRequiredUserId: Long = clientContext.userId.getOrElse(

throw PipelineFailure(BadRequest, "Missing required field: userId"))

/\*\*

\* getOptionalUserId returns a userId if one is set

\*/

def getOptionalUserId: Option[Long] = clientContext.userId

/\*\*

\* getUserIdLoggedOutSupport returns a userId and falls back to 0 if none is set

\*/

def getUserIdLoggedOutSupport: Long = clientContext.userId.getOrElse(0L)

/\*\*

\* getUserOrGuestId returns a userId or a guestId if no userId has been set

\*/

def getUserOrGuestId: Option[Long] = clientContext.userId.orElse(clientContext.guestId)

/\*\*

\* getCountryCode returns a country code if one is set

\*/

def getCountryCode: Option[String] = clientContext.countryCode

/\*\*

\* getLanguageCode returns a language code if one is set

\*/

def getLanguageCode: Option[String] = clientContext.languageCode

/\*\*

\* isLoggedOut returns true if the user is logged out (no userId present).

\*

\* @note this can be useful in conjunction with [[getUserIdLoggedOutSupport]]

\*/

def isLoggedOut: Boolean = clientContext.userId.isEmpty

}