package com.twitter.frigate.pushservice.util

import com.twitter.frigate.common.store.deviceinfo.DeviceInfo

import com.twitter.frigate.common.store.deviceinfo.MobileClientType

import com.twitter.frigate.pushservice.model.PushTypes.Target

import com.twitter.util.Future

import com.twitter.finagle.stats.NullStatsReceiver

import com.twitter.finagle.stats.StatsReceiver

object PushDeviceUtil {

def isPrimaryDeviceAndroid(deviceInfoOpt: Option[DeviceInfo]): Boolean = {

deviceInfoOpt.exists {

\_.guessedPrimaryClient.exists { clientType =>

(clientType == MobileClientType.Android) || (clientType == MobileClientType.AndroidLite)

}

}

}

def isPrimaryDeviceIOS(deviceInfoOpt: Option[DeviceInfo]): Boolean = {

deviceInfoOpt.exists {

\_.guessedPrimaryClient.exists { clientType =>

(clientType == MobileClientType.Iphone) || (clientType == MobileClientType.Ipad)

}

}

}

def isPushRecommendationsEligible(target: Target): Future[Boolean] =

target.deviceInfo.map(\_.exists(\_.isRecommendationsEligible))

def isTopicsEligible(

target: Target,

statsReceiver: StatsReceiver = NullStatsReceiver

): Future[Boolean] = {

val isTopicsSkipFatigue = Future.True

Future.join(isTopicsSkipFatigue, target.deviceInfo.map(\_.exists(\_.isTopicsEligible))).map {

case (isTopicsNotFatigue, isTopicsEligibleSetting) =>

isTopicsNotFatigue && isTopicsEligibleSetting

}

}

def isSpacesEligible(target: Target): Future[Boolean] =

target.deviceInfo.map(\_.exists(\_.isSpacesEligible))

def isNtabOnlyEligible: Future[Boolean] = {

Future.False

}

def isRecommendationsEligible(target: Target): Future[Boolean] = {

Future.join(isPushRecommendationsEligible(target), isNtabOnlyEligible).map {

case (isPushRecommendation, isNtabOnly) => isPushRecommendation || isNtabOnly

case \_ => false

}

}

}