package com.twitter.frigate.pushservice.params

import com.twitter.conversions.DurationOps.\_

import com.twitter.frigate.user\_states.thriftscala.UserState

import java.util.Locale

object PushConstants {

final val ServiceProdEnvironmentName = "prod"

final val RestrictLightRankingCandidatesThreshold = 1

final val DownSampleLightRankingScribeCandidatesRate = 1

final val NewUserLookbackWindow = 1.days

final val PushCapInactiveUserAndroid = 1

final val PushCapInactiveUserIos = 1

final val PushCapLightOccasionalOpenerUserAndroid = 1

final val PushCapLightOccasionalOpenerUserIos = 1

final val UserStateToPushCapIos = Map(

UserState.Inactive.name -> PushCapInactiveUserIos,

UserState.LightOccasionalOpener.name -> PushCapLightOccasionalOpenerUserIos

)

final val UserStateToPushCapAndroid = Map(

UserState.Inactive.name -> PushCapInactiveUserAndroid,

UserState.LightOccasionalOpener.name -> PushCapLightOccasionalOpenerUserAndroid

)

final val AcceptableTimeSinceLastNegativeResponse = 1.days

final val DefaultLookBackForHistory = 1.hours

final val DefaultEventMediaUrl = ""

final val ConnectTabPushTapThrough = "i/connect\_people"

final val AddressBookUploadTapThrough = "i/flow/mr-address-book-upload"

final val InterestPickerTapThrough = "i/flow/mr-interest-picker"

final val CompleteOnboardingInterestAddressTapThrough = "i/flow/mr-interest-address"

final val IndiaCountryCode = "IN"

final val JapanCountryCode = Locale.JAPAN.getCountry.toUpperCase

final val UKCountryCode = Locale.UK.getCountry.toUpperCase

final val IndiaTimeZoneCode = "Asia/Kolkata"

final val JapanTimeZoneCode = "Asia/Tokyo"

final val UKTimeZoneCode = "Europe/London"

final val countryCodeToTimeZoneMap = Map(

IndiaCountryCode -> IndiaTimeZoneCode,

JapanCountryCode -> JapanTimeZoneCode,

UKCountryCode -> UKTimeZoneCode

)

final val AbuseStrike\_Top2Percent\_Id = "AbuseStrike\_Top2Percent\_Id"

final val AbuseStrike\_Top1Percent\_Id = "AbuseStrike\_Top1Percent\_Id"

final val AbuseStrike\_Top05Percent\_Id = "AbuseStrike\_Top05Percent\_Id"

final val AbuseStrike\_Top025Percent\_Id = "AbuseStrike\_Top025Percent\_Id"

final val AllSpamReportsPerFav\_Top1Percent\_Id = "AllSpamReportsPerFav\_Top1Percent\_Id"

final val ReportsPerFav\_Top1Percent\_Id = "ReportsPerFav\_Top1Percent\_Id"

final val ReportsPerFav\_Top2Percent\_Id = "ReportsPerFav\_Top2Percent\_Id"

final val MediaUnderstanding\_Nudity\_Id = "MediaUnderstanding\_Nudity\_Id"

final val MediaUnderstanding\_Beauty\_Id = "MediaUnderstanding\_Beauty\_Id"

final val MediaUnderstanding\_SinglePerson\_Id = "MediaUnderstanding\_SinglePerson\_Id"

final val PornList\_Id = "PornList\_Id"

final val PornographyAndNsfwContent\_Id = "PornographyAndNsfwContent\_Id"

final val SexLife\_Id = "SexLife\_Id"

final val SexLifeOrSexualOrientation\_Id = "SexLifeOrSexualOrientation\_Id"

final val ProfanityFilter\_Id = "ProfanityFilter\_Id"

final val TweetSemanticCoreIdFeature = "tweet.core.tweet.semantic\_core\_annotations"

final val targetUserGenderFeatureName = "Target.User.Gender"

final val targetUserAgeFeatureName = "Target.User.AgeBucket"

final val targetUserPreferredLanguage = "user.language.user.preferred\_contents"

final val tweetAgeInHoursFeatureName = "RecTweet.TweetyPieResult.TweetAgeInHrs"

final val authorActiveFollowerFeatureName = "RecTweetAuthor.User.ActiveFollowers"

final val favFeatureName = "tweet.core.tweet\_counts.favorite\_count"

final val sentFeatureName =

"tweet.magic\_recs\_tweet\_real\_time\_aggregates\_v2.pair.v2.magicrecs.realtime.is\_sent.any\_feature.Duration.Top.count"

final val authorSendCountFeatureName =

"tweet\_author\_aggregate.pair.any\_label.any\_feature.28.days.count"

final val authorReportCountFeatureName =

"tweet\_author\_aggregate.pair.label.reportTweetDone.any\_feature.28.days.count"

final val authorDislikeCountFeatureName =

"tweet\_author\_aggregate.pair.label.ntab.isDisliked.any\_feature.28.days.count"

final val TweetLikesFeatureName = "tweet.core.tweet\_counts.favorite\_count"

final val TweetRepliesFeatureName = "tweet.core.tweet\_counts.reply\_count"

final val EnableCopyFeaturesForIbis2ModelValues = "has\_copy\_features"

final val EmojiFeatureNameForIbis2ModelValues = "emoji"

final val TargetFeatureNameForIbis2ModelValues = "target"

final val CopyBodyExpIbisModelValues = "enable\_body\_exp"

final val TweetMediaEmbeddingBQKeyIds = Seq(

230, 110, 231, 111, 232, 233, 112, 113, 234, 235, 114, 236, 115, 237, 116, 117, 238, 118, 239,

119, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 240, 120, 241, 121, 242, 0, 1, 122, 243, 244, 123,

2, 124, 245, 3, 4, 246, 125, 5, 126, 247, 127, 248, 6, 128, 249, 7, 8, 129, 9, 20, 21, 22, 23,

24, 25, 26, 27, 28, 29, 250, 130, 251, 252, 131, 132, 253, 133, 254, 134, 255, 135, 136, 137,

138, 139, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 140, 141, 142, 143, 144, 145, 146, 147, 148,

149, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159,

50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 60,

61, 62, 63, 64, 65, 66, 67, 68, 69, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 70, 71,

72, 73, 74, 75, 76, 77, 78, 79, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 80, 81, 82,

83, 84, 85, 86, 87, 88, 89, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 90, 91, 92, 93,

94, 95, 96, 97, 98, 99, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213,

214, 215, 216, 217, 218, 219, 220, 100, 221, 101, 222, 223, 102, 224, 103, 104, 225, 105, 226,

227, 106, 107, 228, 108, 229, 109

)

final val SportsEventDomainId = 6L

final val OoncQualityCombinedScore = "OoncQualityCombinedScore"

}

object PushQPSLimitConstants {

final val PerspectiveStoreQPS = 100000

final val IbisOrNTabQPSForRFPH = 100000

final val SocialGraphServiceBatchSize = 100000

}