package com.twitter.follow\_recommendations.common.feature\_hydration.sources

import com.twitter.conversions.DurationOps.\_

import com.twitter.ml.featurestore.catalog.datasets.core.UserMobileSdkDataset

import com.twitter.ml.featurestore.catalog.datasets.core.UsersourceEntityDataset

import com.twitter.ml.featurestore.catalog.datasets.customer\_journey.PostNuxAlgorithmIdAggregateDataset

import com.twitter.ml.featurestore.catalog.datasets.customer\_journey.PostNuxAlgorithmTypeAggregateDataset

import com.twitter.ml.featurestore.catalog.datasets.magicrecs.NotificationSummariesEntityDataset

import com.twitter.ml.featurestore.catalog.datasets.onboarding.MetricCenterUserCountingFeaturesDataset

import com.twitter.ml.featurestore.catalog.datasets.onboarding.UserWtfAlgorithmAggregateFeaturesDataset

import com.twitter.ml.featurestore.catalog.datasets.onboarding.WhoToFollowPostNuxFeaturesDataset

import com.twitter.ml.featurestore.catalog.datasets.rux.UserRecentReactivationTimeDataset

import com.twitter.ml.featurestore.catalog.datasets.timelines.AuthorFeaturesEntityDataset

import com.twitter.ml.featurestore.lib.dataset.DatasetParams

import com.twitter.ml.featurestore.lib.dataset.online.BatchingPolicy

import com.twitter.ml.featurestore.lib.params.FeatureStoreParams

import com.twitter.strato.opcontext.Attribution.ManhattanAppId

import com.twitter.strato.opcontext.ServeWithin

object FeatureStoreParameters {

private val FeatureServiceBatchSize = 100

val featureStoreParams = FeatureStoreParams(

global = DatasetParams(

serveWithin = Some(ServeWithin(duration = 240.millis, roundTripAllowance = None)),

attributions = Seq(

ManhattanAppId("omega", "wtf\_impression\_store"),

ManhattanAppId("athena", "wtf\_athena"),

ManhattanAppId("starbuck", "wtf\_starbuck"),

ManhattanAppId("apollo", "wtf\_apollo")

),

batchingPolicy = Some(BatchingPolicy.Isolated(FeatureServiceBatchSize))

),

perDataset = Map(

MetricCenterUserCountingFeaturesDataset.id ->

DatasetParams(

stratoSuffix = Some("onboarding"),

batchingPolicy = Some(BatchingPolicy.Isolated(200))

),

UsersourceEntityDataset.id ->

DatasetParams(

stratoSuffix = Some("onboarding")

),

WhoToFollowPostNuxFeaturesDataset.id ->

DatasetParams(

stratoSuffix = Some("onboarding"),

batchingPolicy = Some(BatchingPolicy.Isolated(200))

),

AuthorFeaturesEntityDataset.id ->

DatasetParams(

stratoSuffix = Some("onboarding"),

batchingPolicy = Some(BatchingPolicy.Isolated(10))

),

UserRecentReactivationTimeDataset.id -> DatasetParams(

stratoSuffix =

None // removed due to low hit rate. we should use a negative cache in the future

),

UserWtfAlgorithmAggregateFeaturesDataset.id -> DatasetParams(

stratoSuffix = None

),

NotificationSummariesEntityDataset.id -> DatasetParams(

stratoSuffix = Some("onboarding"),

serveWithin = Some(ServeWithin(duration = 45.millis, roundTripAllowance = None)),

batchingPolicy = Some(BatchingPolicy.Isolated(10))

),

UserMobileSdkDataset.id -> DatasetParams(

stratoSuffix = Some("onboarding")

),

PostNuxAlgorithmIdAggregateDataset.id -> DatasetParams(

stratoSuffix = Some("onboarding")

),

PostNuxAlgorithmTypeAggregateDataset.id -> DatasetParams(

stratoSuffix = Some("onboarding")

),

),

enableFeatureGenerationStats = true

)

}