package com.twitter.timelines.prediction.common.aggregates.real\_time

import com.twitter.finagle.mtls.authentication.ServiceIdentifier

import com.twitter.finagle.stats.StatsReceiver

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

import com.twitter.ml.featurestore.catalog.datasets.geo.GeoUserLocationDataset

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

import com.twitter.ml.featurestore.lib.export.strato.FeatureStoreAppNames

import com.twitter.ml.featurestore.lib.online.FeatureStoreClient

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

import com.twitter.strato.client.{Client, Strato}

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

import com.twitter.util.Duration

private[real\_time] object FeatureStoreUtils {

private def mkStratoClient(serviceIdentifier: ServiceIdentifier): Client =

Strato.client

.withMutualTls(serviceIdentifier)

.withRequestTimeout(Duration.fromMilliseconds(50))

.build()

private val featureStoreParams: FeatureStoreParams =

FeatureStoreParams(

perDataset = Map(

UserFeaturesDataset.id ->

DatasetParams(

stratoSuffix = Some(FeatureStoreAppNames.Timelines),

attributions = Seq(ManhattanAppId("athena", "timelines\_aggregates\_v2\_features\_by\_user"))

),

GeoUserLocationDataset.id ->

DatasetParams(

attributions = Seq(ManhattanAppId("starbuck", "timelines\_geo\_features\_by\_user"))

)

)

)

def mkFeatureStoreClient(

serviceIdentifier: ServiceIdentifier,

statsReceiver: StatsReceiver

): FeatureStoreClient = {

com.twitter.server.Init() // necessary in order to use WilyNS path

val stratoClient: Client = mkStratoClient(serviceIdentifier)

val featureStoreClient: FeatureStoreClient = FeatureStoreClient(

featureSet =

UserFeaturesAdapter.UserFeaturesSet ++ AuthorFeaturesAdapter.UserFeaturesSet ++ TweetFeaturesAdapter.TweetFeaturesSet,

client = stratoClient,

statsReceiver = statsReceiver,

featureStoreParams = featureStoreParams

)

featureStoreClient

}

}