package com.twitter.home\_mixer.product.scored\_tweets.feature\_hydrator.offline\_aggregates

import com.twitter.home\_mixer.param.HomeMixerInjectionNames.TimelineAggregateMetadataRepository

import com.twitter.home\_mixer.param.HomeMixerInjectionNames.TimelineAggregatePartARepository

import com.twitter.product\_mixer.core.model.common.identifier.FeatureHydratorIdentifier

import com.twitter.servo.repository.Repository

import com.twitter.timelines.data\_processing.jobs.timeline\_ranking\_user\_features.TimelinesPartAStoreRegister

import com.twitter.timelines.data\_processing.ml\_util.aggregation\_framework.StoreConfig

import com.twitter.timelines.suggests.common.dense\_data\_record.thriftscala.DenseFeatureMetadata

import com.twitter.user\_session\_store.thriftjava.UserSession

import javax.inject.Inject

import javax.inject.Named

import javax.inject.Singleton

object PartAAggregateRootFeature extends BaseAggregateRootFeature {

override val aggregateStores: Set[StoreConfig[\_]] = TimelinesPartAStoreRegister.allStores

}

@Singleton

class PartAAggregateQueryFeatureHydrator @Inject() (

@Named(TimelineAggregatePartARepository)

repository: Repository[Long, Option[UserSession]],

@Named(TimelineAggregateMetadataRepository)

metadataRepository: Repository[Int, Option[DenseFeatureMetadata]])

extends BaseAggregateQueryFeatureHydrator(

repository,

metadataRepository,

PartAAggregateRootFeature

) {

override val identifier: FeatureHydratorIdentifier =

FeatureHydratorIdentifier("PartAAggregateQuery")

override val features = Set(PartAAggregateRootFeature)

}