package com.twitter.cr\_mixer.module.similarity\_engine

import com.google.inject.Provides

import com.twitter.cr\_mixer.config.TimeoutConfig

import com.twitter.cr\_mixer.model.ModuleNames

import com.twitter.cr\_mixer.model.TweetWithScore

import com.twitter.cr\_mixer.param.decider.CrMixerDecider

import com.twitter.cr\_mixer.param.decider.DeciderConstants

import com.twitter.cr\_mixer.similarity\_engine.ConsumersBasedUserAdGraphSimilarityEngine

import com.twitter.cr\_mixer.similarity\_engine.SimilarityEngine.DeciderConfig

import com.twitter.cr\_mixer.similarity\_engine.SimilarityEngine.GatingConfig

import com.twitter.cr\_mixer.similarity\_engine.SimilarityEngine.SimilarityEngineConfig

import com.twitter.cr\_mixer.similarity\_engine.StandardSimilarityEngine

import com.twitter.cr\_mixer.thriftscala.SimilarityEngineType

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import com.twitter.recos.user\_ad\_graph.thriftscala.ConsumersBasedRelatedAdRequest

import com.twitter.recos.user\_ad\_graph.thriftscala.RelatedAdResponse

import com.twitter.storehaus.ReadableStore

import javax.inject.Named

import javax.inject.Singleton

object ConsumersBasedUserAdGraphSimilarityEngineModule extends TwitterModule {

@Provides

@Singleton

@Named(ModuleNames.ConsumersBasedUserAdGraphSimilarityEngine)

def providesConsumersBasedUserAdGraphSimilarityEngine(

@Named(ModuleNames.ConsumerBasedUserAdGraphStore)

consumersBasedUserAdGraphStore: ReadableStore[

ConsumersBasedRelatedAdRequest,

RelatedAdResponse

],

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

decider: CrMixerDecider

): StandardSimilarityEngine[

ConsumersBasedUserAdGraphSimilarityEngine.Query,

TweetWithScore

] = {

new StandardSimilarityEngine[

ConsumersBasedUserAdGraphSimilarityEngine.Query,

TweetWithScore

](

implementingStore =

ConsumersBasedUserAdGraphSimilarityEngine(consumersBasedUserAdGraphStore, statsReceiver),

identifier = SimilarityEngineType.ConsumersBasedUserTweetGraph,

globalStats = statsReceiver,

engineConfig = SimilarityEngineConfig(

timeout = timeoutConfig.similarityEngineTimeout,

gatingConfig = GatingConfig(

deciderConfig =

Some(DeciderConfig(decider, DeciderConstants.enableUserTweetGraphTrafficDeciderKey)),

enableFeatureSwitch = None

)

),

memCacheConfig = None

)

}

}