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.ModelConfig

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

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

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

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.similarity\_engine.TripEngineQuery

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

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.hermit.store.common.ObservedReadableStore

import com.twitter.inject.TwitterModule

import com.twitter.simclusters\_v2.common.SimClustersEmbedding

import com.twitter.simclusters\_v2.common.UserId

import com.twitter.storehaus.ReadableStore

import com.twitter.trends.trip\_v1.trip\_tweets.thriftscala.TripTweet

import com.twitter.trends.trip\_v1.trip\_tweets.thriftscala.TripDomain

import javax.inject.Named

object ConsumerEmbeddingBasedTripSimilarityEngineModule extends TwitterModule {

@Provides

@Named(ModuleNames.ConsumerEmbeddingBasedTripSimilarityEngine)

def providesConsumerEmbeddingBasedTripSimilarityEngineModule(

@Named(ModuleNames.RmsUserLogFavInterestedInEmbeddingStore)

userLogFavInterestedInEmbeddingStore: ReadableStore[UserId, SimClustersEmbedding],

@Named(ModuleNames.RmsUserFollowInterestedInEmbeddingStore)

userFollowInterestedInEmbeddingStore: ReadableStore[UserId, SimClustersEmbedding],

@Named(ModuleNames.TripCandidateStore)

tripCandidateStore: ReadableStore[TripDomain, Seq[TripTweet]],

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

): StandardSimilarityEngine[TripEngineQuery, TripTweetWithScore] = {

val underlyingStore = ObservedReadableStore(

ConsumerEmbeddingBasedTripSimilarityEngine(

embeddingStoreLookUpMap = Map(

ModelConfig.ConsumerLogFavBasedInterestedInEmbedding -> userLogFavInterestedInEmbeddingStore,

ModelConfig.ConsumerFollowBasedInterestedInEmbedding -> userFollowInterestedInEmbeddingStore,

),

tripCandidateSource = tripCandidateStore,

statsReceiver

))(statsReceiver.scope("TripSimilarityEngine"))

new StandardSimilarityEngine[TripEngineQuery, TripTweetWithScore](

implementingStore = underlyingStore,

identifier = SimilarityEngineType.ExploreTripOfflineSimClustersTweets,

globalStats = statsReceiver,

engineConfig = SimilarityEngineConfig(

timeout = timeoutConfig.similarityEngineTimeout,

gatingConfig = GatingConfig(

deciderConfig = None,

enableFeatureSwitch = None

)

)

)

}

}