package com.twitter.cr\_mixer.module

package similarity\_engine

import com.google.inject.Provides

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

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

import com.twitter.simclusters\_v2.thriftscala.TweetsWithScore

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

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

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

import com.twitter.cr\_mixer.similarity\_engine.DiffusionBasedSimilarityEngine.Query

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

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

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

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

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import com.twitter.storehaus.ReadableStore

import javax.inject.Named

import javax.inject.Singleton

object DiffusionBasedSimilarityEngineModule extends TwitterModule {

@Provides

@Singleton

@Named(ModuleNames.DiffusionBasedSimilarityEngine)

def providesDiffusionBasedSimilarityEngineModule(

@Named(ModuleNames.RetweetBasedDiffusionRecsMhStore)

retweetBasedDiffusionRecsMhStore: ReadableStore[Long, TweetsWithScore],

timeoutConfig: TimeoutConfig,

globalStats: StatsReceiver

): LookupSimilarityEngine[Query, TweetWithScore] = {

val versionedStoreMap = Map(

ModelConfig.RetweetBasedDiffusion -> DiffusionBasedSimilarityEngine(

retweetBasedDiffusionRecsMhStore,

globalStats),

)

new LookupSimilarityEngine[Query, TweetWithScore](

versionedStoreMap = versionedStoreMap,

identifier = SimilarityEngineType.DiffusionBasedTweet,

globalStats = globalStats,

engineConfig = SimilarityEngineConfig(

timeout = timeoutConfig.similarityEngineTimeout,

gatingConfig = GatingConfig(

deciderConfig = None,

enableFeatureSwitch = None

)

)

)

}

}