package com.twitter.cr\_mixer.module

package similarity\_engine

import com.google.inject.Provides

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

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

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

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

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

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

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

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.TwhinCollabFilterSimilarityEngine

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

/\*\*

\* TwhinCandidatesLookupSimilarityEngineModule routes the request to the corresponding

\* twhin based candidate store which follow the same pattern as TwHIN Collaborative Filtering.

\*/

object TwhinCollabFilterLookupSimilarityEngineModule extends TwitterModule {

@Provides

@Singleton

@Named(ModuleNames.TwhinCollabFilterSimilarityEngine)

def providesTwhinCollabFilterLookupSimilarityEngineModule(

@Named(ModuleNames.TwhinCollabFilterStratoStoreForFollow)

twhinCollabFilterStratoStoreForFollow: ReadableStore[Long, Seq[TweetId]],

@Named(ModuleNames.TwhinCollabFilterStratoStoreForEngagement)

twhinCollabFilterStratoStoreForEngagement: ReadableStore[Long, Seq[TweetId]],

@Named(ModuleNames.TwhinMultiClusterStratoStoreForFollow)

twhinMultiClusterStratoStoreForFollow: ReadableStore[Long, Seq[TweetId]],

@Named(ModuleNames.TwhinMultiClusterStratoStoreForEngagement)

twhinMultiClusterStratoStoreForEngagement: ReadableStore[Long, Seq[TweetId]],

timeoutConfig: TimeoutConfig,

globalStats: StatsReceiver

): LookupSimilarityEngine[Query, TweetWithScore] = {

val versionedStoreMap = Map(

ModelConfig.TwhinCollabFilterForFollow -> TwhinCollabFilterSimilarityEngine(

twhinCollabFilterStratoStoreForFollow,

globalStats),

ModelConfig.TwhinCollabFilterForEngagement -> TwhinCollabFilterSimilarityEngine(

twhinCollabFilterStratoStoreForEngagement,

globalStats),

ModelConfig.TwhinMultiClusterForFollow -> TwhinCollabFilterSimilarityEngine(

twhinMultiClusterStratoStoreForFollow,

globalStats),

ModelConfig.TwhinMultiClusterForEngagement -> TwhinCollabFilterSimilarityEngine(

twhinMultiClusterStratoStoreForEngagement,

globalStats),

)

new LookupSimilarityEngine[Query, TweetWithScore](

versionedStoreMap = versionedStoreMap,

identifier = SimilarityEngineType.TwhinCollabFilter,

globalStats = globalStats,

engineConfig = SimilarityEngineConfig(

timeout = timeoutConfig.similarityEngineTimeout,

gatingConfig = GatingConfig(

deciderConfig = None,

enableFeatureSwitch = None

)

)

)

}

}