package com.twitter.tsp.modules

import com.google.inject.Module

import com.google.inject.Provides

import com.google.inject.Singleton

import com.twitter.app.Flag

import com.twitter.bijection.scrooge.BinaryScalaCodec

import com.twitter.conversions.DurationOps.\_

import com.twitter.finagle.memcached.{Client => MemClient}

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.hermit.store.common.ObservedMemcachedReadableStore

import com.twitter.inject.TwitterModule

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

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

import com.twitter.storehaus.ReadableStore

import com.twitter.strato.client.{Client => StratoClient}

import com.twitter.tsp.stores.RepresentationScorerStore

object RepresentationScorerStoreModule extends TwitterModule {

override def modules: Seq[Module] = Seq(UnifiedCacheClient)

private val tspRepresentationScoringColumnPath: Flag[String] = flag[String](

name = "tsp.representationScoringColumnPath",

default = "recommendations/representation\_scorer/score",

help = "Strato column path for Representation Scorer Store"

)

@Provides

@Singleton

def providesRepresentationScorerStore(

statsReceiver: StatsReceiver,

stratoClient: StratoClient,

tspUnifiedCacheClient: MemClient

): ReadableStore[ScoreId, Score] = {

val underlyingStore =

RepresentationScorerStore(stratoClient, tspRepresentationScoringColumnPath(), statsReceiver)

ObservedMemcachedReadableStore.fromCacheClient(

backingStore = underlyingStore,

cacheClient = tspUnifiedCacheClient,

ttl = 2.hours

)(

valueInjection = BinaryScalaCodec(Score),

statsReceiver = statsReceiver.scope("RepresentationScorerStore"),

keyToString = { k: ScoreId => s"rsx/$k" }

)

}

}