package com.twitter.cr\_mixer.module

import com.google.inject.Provides

import com.google.inject.Singleton

import com.twitter.bijection.scrooge.BinaryScalaCodec

import com.twitter.conversions.DurationOps.\_

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

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

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

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import com.twitter.hermit.store.common.ReadableWritableStore

import com.twitter.hermit.store.common.ObservedReadableWritableMemcacheStore

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

import javax.inject.Named

object TweetRecommendationResultsStoreModule extends TwitterModule {

@Provides

@Singleton

def providesTweetRecommendationResultsStore(

@Named(ModuleNames.TweetRecommendationResultsCache) tweetRecommendationResultsCacheClient: MemcachedClient,

statsReceiver: StatsReceiver

): ReadableWritableStore[UserId, CrMixerTweetResponse] = {

ObservedReadableWritableMemcacheStore.fromCacheClient(

cacheClient = tweetRecommendationResultsCacheClient,

ttl = 24.hours)(

valueInjection = BinaryScalaCodec(CrMixerTweetResponse),

statsReceiver = statsReceiver.scope("TweetRecommendationResultsMemcacheStore"),

keyToString = { k: UserId => k.toString }

)

}

}