package com.twitter.timelineranker.clients

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

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.logging.Logger

import com.twitter.servo.cache.FinagleMemcache

import com.twitter.servo.cache.MemcacheCache

import com.twitter.servo.cache.ObservableMemcache

import com.twitter.servo.cache.Serializer

import com.twitter.servo.cache.StatsReceiverCacheObserver

import com.twitter.timelines.util.stats.RequestScope

import com.twitter.timelines.util.stats.ScopedFactory

import com.twitter.util.Duration

/\*\*

\* Factory to create a servo Memcache-backed Cache object. Clients are required to provide a

\* serializer/deserializer for keys and values.

\*/

class MemcacheFactory(memcacheClient: FinagleMemcacheClient, statsReceiver: StatsReceiver) {

private[this] val logger = Logger.get(getClass.getSimpleName)

def apply[K, V](

keySerializer: K => String,

valueSerializer: Serializer[V],

ttl: Duration

): MemcacheCache[K, V] = {

new MemcacheCache[K, V](

memcache = new ObservableMemcache(

new FinagleMemcache(memcacheClient),

new StatsReceiverCacheObserver(statsReceiver, 1000, logger)

),

ttl = ttl,

serializer = valueSerializer,

transformKey = keySerializer

)

}

}

class ScopedMemcacheFactory(memcacheClient: FinagleMemcacheClient, statsReceiver: StatsReceiver)

extends ScopedFactory[MemcacheFactory] {

override def scope(scope: RequestScope): MemcacheFactory = {

new MemcacheFactory(

memcacheClient,

statsReceiver.scope("memcache", scope.scope)

)

}

}