package com.twitter.servo.cache

import com.twitter.util.{Duration, Future}

trait CounterCache[K] extends Cache[K, Long] {

def incr(key: K, delta: Int = 1): Future[Option[Long]]

def decr(key: K, delta: Int = 1): Future[Option[Long]]

}

class MemcacheCounterCache[K](

memcache: Memcache,

ttl: Duration,

transformKey: KeyTransformer[K] = ((k: K) => k.toString): (K => java.lang.String))

extends MemcacheCache[K, Long](memcache, ttl, CounterSerializer, transformKey)

with CounterCache[K]

class NullCounterCache[K] extends NullCache[K, Long] with CounterCache[K] {

override def incr(key: K, delta: Int = 1): Future[Option[Long]] = Future.value(Some(0L))

override def decr(key: K, delta: Int = 1): Future[Option[Long]] = Future.value(Some(0L))

}