package com.twitter.tweetypie.serverutil

import com.twitter.finagle.memcached

import com.twitter.finagle.memcached.CasResult

import com.twitter.io.Buf

import com.twitter.tweetypie.Future

import com.twitter.tweetypie.Time

import java.lang

/\*\*

\* This will be used during CI test runs, in the no-cache scenarios for both DCs.

\* We are treating this as cache of instantaneous expiry. MockClient uses an in-memory map as

\* an underlying data-store, we extend it and prevent any writes to the map - thus making sure

\* it's always empty.

\*/

class NullMemcacheClient extends memcached.MockClient {

override def set(key: String, flags: Int, expiry: Time, value: Buf): Future[Unit] = Future.Done

override def add(key: String, flags: Int, expiry: Time, value: Buf): Future[lang.Boolean] =

Future.value(true)

override def append(key: String, flags: Int, expiry: Time, value: Buf): Future[lang.Boolean] =

Future.value(false)

override def prepend(key: String, flags: Int, expiry: Time, value: Buf): Future[lang.Boolean] =

Future.value(false)

override def replace(key: String, flags: Int, expiry: Time, value: Buf): Future[lang.Boolean] =

Future.value(false)

override def checkAndSet(

key: String,

flags: Int,

expiry: Time,

value: Buf,

casUnique: Buf

): Future[CasResult] = Future.value(CasResult.NotFound)

override def delete(key: String): Future[lang.Boolean] = Future.value(false)

override def incr(key: String, delta: Long): Future[Option[lang.Long]] =

Future.value(None)

override def decr(key: String, delta: Long): Future[Option[lang.Long]] =

Future.value(None)

}