package com.twitter.representationscorer.modules

import com.google.inject.Provides

import com.twitter.finagle.memcached.Client

import javax.inject.Singleton

import com.twitter.conversions.DurationOps.\_

import com.twitter.inject.TwitterModule

import com.twitter.finagle.mtls.authentication.ServiceIdentifier

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.storehaus\_internal.memcache.MemcacheStore

import com.twitter.storehaus\_internal.util.ClientName

import com.twitter.storehaus\_internal.util.ZkEndPoint

object CacheModule extends TwitterModule {

private val cacheDest = flag[String]("cache\_module.dest", "Path to memcache service")

private val timeout = flag[Int]("memcache.timeout", "Memcache client timeout")

private val retries = flag[Int]("memcache.retries", "Memcache timeout retries")

@Singleton

@Provides

def providesCache(

serviceIdentifier: ServiceIdentifier,

stats: StatsReceiver

): Client =

MemcacheStore.memcachedClient(

name = ClientName("memcache\_representation\_manager"),

dest = ZkEndPoint(cacheDest()),

timeout = timeout().milliseconds,

retries = retries(),

statsReceiver = stats.scope("cache\_client"),

serviceIdentifier = serviceIdentifier

)

}