package com.twitter.home\_mixer.functional\_component.candidate\_source

import com.google.inject.name.Named

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

import com.twitter.home\_mixer.param.HomeMixerInjectionNames.StaleTweetsCache

import com.twitter.product\_mixer.core.functional\_component.candidate\_source.CandidateSource

import com.twitter.product\_mixer.core.model.common.identifier.CandidateSourceIdentifier

import com.twitter.stitch.Stitch

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class StaleTweetsCacheCandidateSource @Inject() (

@Named(StaleTweetsCache) staleTweetsCache: MemcachedClient)

extends CandidateSource[Seq[Long], Long] {

override val identifier: CandidateSourceIdentifier = CandidateSourceIdentifier("StaleTweetsCache")

private val StaleTweetsCacheKeyPrefix = "v1\_"

override def apply(request: Seq[Long]): Stitch[Seq[Long]] = {

val keys = request.map(StaleTweetsCacheKeyPrefix + \_)

Stitch.callFuture(staleTweetsCache.get(keys).map { tweets =>

tweets.map {

case (k, \_) => k.replaceFirst(StaleTweetsCacheKeyPrefix, "").toLong

}.toSeq

})

}

}