package com.twitter.follow\_recommendations.common.candidate\_sources.base

import com.twitter.escherbird.util.stitchcache.StitchCache

import com.twitter.finagle.stats.StatsReceiver

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 com.twitter.util.Duration

class CachedCandidateSource[K <: Object, V <: Object](

candidateSource: CandidateSource[K, V],

maxCacheSize: Int,

cacheTTL: Duration,

statsReceiver: StatsReceiver,

override val identifier: CandidateSourceIdentifier)

extends CandidateSource[K, V] {

private val cache = StitchCache[K, Seq[V]](

maxCacheSize = maxCacheSize,

ttl = cacheTTL,

statsReceiver = statsReceiver.scope(identifier.name, "cache"),

underlyingCall = (k: K) => candidateSource(k)

)

override def apply(target: K): Stitch[Seq[V]] = cache.readThrough(target)

}