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

import com.google.inject.Singleton

import com.google.inject.name.Named

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.follow\_recommendations.common.constants.GuiceNamedConstants

import com.twitter.hermit.candidate.thriftscala.Candidates

import com.twitter.hermit.model.Algorithm

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

import com.twitter.strato.client.Fetcher

import com.twitter.util.Duration

import javax.inject.Inject

@Singleton

class DBV2SimsStore @Inject() (

@Named(GuiceNamedConstants.DBV2\_SIMS\_FETCHER) fetcher: Fetcher[Long, Unit, Candidates])

extends StratoBasedSimsCandidateSourceWithUnitView(

fetcher,

identifier = DBV2SimsStore.Identifier)

@Singleton

class CachedDBV2SimsStore @Inject() (

@Named(GuiceNamedConstants.DBV2\_SIMS\_FETCHER) fetcher: Fetcher[Long, Unit, Candidates],

statsReceiver: StatsReceiver)

extends CacheBasedSimsStore(

id = DBV2SimsStore.Identifier,

fetcher = fetcher,

maxCacheSize = DBV2SimsStore.MaxCacheSize,

cacheTtl = DBV2SimsStore.CacheTTL,

statsReceiver = statsReceiver.scope("CachedDBV2SimsStore", "cache")

)

object DBV2SimsStore {

val Identifier = CandidateSourceIdentifier(Algorithm.Sims.toString)

val MaxCacheSize = 1000

val CacheTTL: Duration = Duration.fromHours(24)

}