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

import com.google.inject.Singleton

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.hermit.model.Algorithm

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

import com.twitter.strato.generated.client.onboarding.userrecs.NewSimsRefreshOnUserClientColumn

import com.twitter.util.Duration

import javax.inject.Inject

@Singleton

class DBV2SimsRefreshStore @Inject() (

newSimsRefreshOnUserClientColumn: NewSimsRefreshOnUserClientColumn)

extends StratoBasedSimsCandidateSourceWithUnitView(

fetcher = newSimsRefreshOnUserClientColumn.fetcher,

identifier = DBV2SimsRefreshStore.Identifier)

@Singleton

class CachedDBV2SimsRefreshStore @Inject() (

newSimsRefreshOnUserClientColumn: NewSimsRefreshOnUserClientColumn,

statsReceiver: StatsReceiver)

extends CacheBasedSimsStore(

id = DBV2SimsRefreshStore.Identifier,

fetcher = newSimsRefreshOnUserClientColumn.fetcher,

maxCacheSize = DBV2SimsRefreshStore.MaxCacheSize,

cacheTtl = DBV2SimsRefreshStore.CacheTTL,

statsReceiver = statsReceiver.scope("CachedDBV2SimsRefreshStore", "cache")

)

object DBV2SimsRefreshStore {

val Identifier = CandidateSourceIdentifier(Algorithm.Sims.toString)

val MaxCacheSize = 5000

val CacheTTL: Duration = Duration.fromHours(24)

}