package com.twitter.simclusters\_v2.summingbird.stores

import com.twitter.simclusters\_v2.summingbird.common.EntityUtil

import com.twitter.simclusters\_v2.thriftscala.\_

import com.twitter.storehaus.ReadableStore

import com.twitter.util.Future

import com.twitter.util.Time

case class TopKClustersForEntityReadableStore(

underlyingStore: ReadableStore[EntityWithVersion, TopKClustersWithScores])

extends ReadableStore[EntityWithVersion, TopKClustersWithScores] {

override def multiGet[K1 <: EntityWithVersion](

ks: Set[K1]

): Map[K1, Future[Option[TopKClustersWithScores]]] = {

val nowInMs = Time.now.inMilliseconds

underlyingStore

.multiGet(ks)

.mapValues { resFuture =>

resFuture.map { resOpt =>

resOpt.map { clustersWithScores =>

clustersWithScores.copy(

topClustersByFavClusterNormalizedScore = EntityUtil.updateScoreWithLatestTimestamp(

clustersWithScores.topClustersByFavClusterNormalizedScore,

nowInMs

),

topClustersByFollowClusterNormalizedScore = EntityUtil.updateScoreWithLatestTimestamp(

clustersWithScores.topClustersByFollowClusterNormalizedScore,

nowInMs

)

)

}

}

}

}

}