package com.twitter.follow\_recommendations.common.candidate\_sources.two\_hop\_random\_walk

import com.twitter.follow\_recommendations.common.candidate\_sources.base.StratoFetcherWithUnitViewSource

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

import com.twitter.follow\_recommendations.common.models.CandidateUser

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.wtf.candidate.thriftscala.{CandidateSeq => TCandidateSeq}

import javax.inject.Inject

import javax.inject.Named

import javax.inject.Singleton

@Singleton

class TwoHopRandomWalkSource @Inject() (

@Named(GuiceNamedConstants.TWO\_HOP\_RANDOM\_WALK\_FETCHER) fetcher: Fetcher[

Long,

Unit,

TCandidateSeq

]) extends StratoFetcherWithUnitViewSource[Long, TCandidateSeq](

fetcher,

TwoHopRandomWalkSource.Identifier) {

override def map(targetUserId: Long, tCandidateSeq: TCandidateSeq): Seq[CandidateUser] =

TwoHopRandomWalkSource.map(targetUserId, tCandidateSeq)

}

object TwoHopRandomWalkSource {

def map(targetUserId: Long, tCandidateSeq: TCandidateSeq): Seq[CandidateUser] = {

tCandidateSeq.candidates

.sortBy(-\_.score)

.map { tCandidate =>

CandidateUser(id = tCandidate.userId, score = Some(tCandidate.score))

}

}

val Identifier: CandidateSourceIdentifier =

CandidateSourceIdentifier(Algorithm.TwoHopRandomWalk.toString)

}