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

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

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

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

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

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

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

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

import com.twitter.strato.client.Fetcher

abstract class StratoBasedSimsCandidateSource[U](

fetcher: Fetcher[Long, U, Candidates],

view: U,

override val identifier: CandidateSourceIdentifier)

extends StratoFetcherSource[Long, U, Candidates](fetcher, view, identifier) {

override def map(target: Long, candidates: Candidates): Seq[CandidateUser] =

StratoBasedSimsCandidateSource.map(target, candidates)

}

object StratoBasedSimsCandidateSource {

def map(target: Long, candidates: Candidates): Seq[CandidateUser] = {

for {

candidate <- candidates.candidates

} yield CandidateUser(

id = candidate.userId,

score = Some(candidate.score),

reason = Some(

Reason(

Some(

AccountProof(

similarToProof = Some(SimilarToProof(Seq(target)))

)

)

)

)

)

}

}