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

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

import com.twitter.product\_mixer.core.functional\_component.candidate\_source.CandidateSource

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

import com.twitter.stitch.Stitch

import com.twitter.strato.client.Fetcher

abstract class StratoFetcherSource[K, U, V](

fetcher: Fetcher[K, U, V],

view: U,

override val identifier: CandidateSourceIdentifier)

extends CandidateSource[K, CandidateUser] {

def map(user: K, v: V): Seq[CandidateUser]

override def apply(target: K): Stitch[Seq[CandidateUser]] = {

fetcher

.fetch(target, view)

.map { result =>

result.v

.map { candidates => map(target, candidates) }

.getOrElse(Nil)

.map(\_.withCandidateSource(identifier))

}

}

}