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

import com.google.inject.Singleton

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

import com.twitter.hermit.model.Algorithm

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

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

import com.twitter.product\_mixer.core.model.marshalling.request.HasClientContext

import com.twitter.stitch.Stitch

import com.twitter.timelines.configapi.HasParams

import javax.inject.Inject

@Singleton

class PopCountryBackFillSource @Inject() (popGeoSource: PopGeoSource)

extends CandidateSource[HasClientContext with HasParams, CandidateUser] {

override val identifier: CandidateSourceIdentifier = PopCountryBackFillSource.Identifier

override def apply(target: HasClientContext with HasParams): Stitch[Seq[CandidateUser]] = {

target.getOptionalUserId

.map(\_ =>

popGeoSource(PopCountryBackFillSource.DefaultKey)

.map(\_.take(PopCountryBackFillSource.MaxResults).map(\_.withCandidateSource(identifier))))

.getOrElse(Stitch.Nil)

}

}

object PopCountryBackFillSource {

val Identifier: CandidateSourceIdentifier =

CandidateSourceIdentifier(Algorithm.PopCountryBackFill.toString)

val MaxResults = 40

val DefaultKey = "country\_US"

}