package com.twitter.product\_mixer.component\_library.candidate\_source.timeline\_ranker

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.timelineranker.{thriftscala => t}

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class TimelineRankerRecapCandidateSource @Inject() (

timelineRankerClient: t.TimelineRanker.MethodPerEndpoint)

extends CandidateSource[t.RecapQuery, t.CandidateTweet] {

override val identifier: CandidateSourceIdentifier =

CandidateSourceIdentifier("TimelineRankerRecap")

override def apply(

request: t.RecapQuery

): Stitch[Seq[t.CandidateTweet]] = {

Stitch

.callFuture(timelineRankerClient.getRecapCandidatesFromAuthors(Seq(request)))

.map { response: Seq[t.GetCandidateTweetsResponse] =>

response.headOption.flatMap(\_.candidates).getOrElse(Seq.empty).filter(\_.tweet.nonEmpty)

}

}

}