package com.twitter.home\_mixer.product.scored\_tweets.candidate\_source

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.stitch.timelineservice.TimelineService

import com.twitter.timelineservice.{thriftscala => tls}

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class ListsCandidateSource @Inject() (timelineService: TimelineService)

extends CandidateSource[Seq[tls.TimelineQuery], tls.Tweet] {

override val identifier: CandidateSourceIdentifier = CandidateSourceIdentifier("Lists")

override def apply(requests: Seq[tls.TimelineQuery]): Stitch[Seq[tls.Tweet]] = {

val timelines = Stitch.traverse(requests) { request => timelineService.getTimeline(request) }

timelines.map {

\_.flatMap {

\_.entries.collect { case tls.TimelineEntry.Tweet(tweet) => tweet }

}

}

}

}