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

import com.twitter.product\_mixer.core.feature.Feature

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMapBuilder

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

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

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

import com.twitter.product\_mixer.core.pipeline.PipelineQuery

import com.twitter.product\_mixer.core.pipeline.pipeline\_failure.PipelineFailure

import com.twitter.product\_mixer.core.pipeline.pipeline\_failure.UnexpectedCandidateResult

import com.twitter.stitch.Stitch

import com.twitter.timelineranker.{thriftscala => t}

import javax.inject.Inject

import javax.inject.Singleton

/\*\*

\* Source tweets of retweets present in Timeline Ranker candidates list.

\* These tweets are used only for further ranking. They are not returned to the end user.

\*/

case object TimelineRankerUtegSourceTweetsFeature

extends Feature[PipelineQuery, Seq[t.CandidateTweet]]

@Singleton

class TimelineRankerUtegCandidateSource @Inject() (

timelineRankerClient: t.TimelineRanker.MethodPerEndpoint)

extends CandidateSourceWithExtractedFeatures[t.UtegLikedByTweetsQuery, t.CandidateTweet] {

override val identifier: CandidateSourceIdentifier =

CandidateSourceIdentifier("TimelineRankerUteg")

override def apply(

request: t.UtegLikedByTweetsQuery

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

Stitch

.callFuture(timelineRankerClient.getUtegLikedByTweetCandidates(Seq(request)))

.map { response =>

val result = response.headOption.getOrElse(

throw PipelineFailure(UnexpectedCandidateResult, "Empty Timeline Ranker response"))

val candidates = result.candidates.toSeq.flatten

val sourceTweets = result.sourceTweets.toSeq.flatten

val candidateSourceFeatures = FeatureMapBuilder()

.add(TimelineRankerUtegSourceTweetsFeature, sourceTweets)

.build()

CandidatesWithSourceFeatures(candidates = candidates, features = candidateSourceFeatures)

}

}

}