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

import com.twitter.home\_mixer.product.scored\_tweets.feature\_hydrator.TweetypieStaticEntitiesFeatureHydrator

import com.twitter.home\_mixer.functional\_component.filter.ReplyFilter

import com.twitter.home\_mixer.functional\_component.filter.RetweetFilter

import com.twitter.home\_mixer.product.scored\_tweets.candidate\_source.ListsCandidateSource

import com.twitter.home\_mixer.product.scored\_tweets.feature\_hydrator.ListIdsFeature

import com.twitter.home\_mixer.product.scored\_tweets.gate.MinCachedTweetsGate

import com.twitter.home\_mixer.product.scored\_tweets.model.ScoredTweetsQuery

import com.twitter.home\_mixer.product.scored\_tweets.param.ScoredTweetsParam.CachedScoredTweets

import com.twitter.home\_mixer.product.scored\_tweets.param.ScoredTweetsParam.CandidatePipeline

import com.twitter.home\_mixer.product.scored\_tweets.response\_transformer.ScoredTweetsListsResponseFeatureTransformer

import com.twitter.product\_mixer.component\_library.gate.NonEmptySeqFeatureGate

import com.twitter.product\_mixer.component\_library.model.candidate.TweetCandidate

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

import com.twitter.product\_mixer.core.functional\_component.feature\_hydrator.BaseCandidateFeatureHydrator

import com.twitter.product\_mixer.core.functional\_component.filter.Filter

import com.twitter.product\_mixer.core.functional\_component.gate.Gate

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidateFeatureTransformer

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidatePipelineQueryTransformer

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidatePipelineResultsTransformer

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

import com.twitter.product\_mixer.core.pipeline.candidate.CandidatePipelineConfig

import com.twitter.timelines.configapi.decider.DeciderParam

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

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class ScoredTweetsListsCandidatePipelineConfig @Inject() (

listsCandidateSource: ListsCandidateSource,

tweetypieStaticEntitiesHydrator: TweetypieStaticEntitiesFeatureHydrator)

extends CandidatePipelineConfig[

ScoredTweetsQuery,

Seq[t.TimelineQuery],

t.Tweet,

TweetCandidate

] {

override val identifier: CandidatePipelineIdentifier =

CandidatePipelineIdentifier("ScoredTweetsLists")

private val MaxTweetsToFetchPerList = 20

override val enabledDeciderParam: Option[DeciderParam[Boolean]] =

Some(CandidatePipeline.EnableListsParam)

override val gates: Seq[Gate[ScoredTweetsQuery]] = Seq(

NonEmptySeqFeatureGate(ListIdsFeature),

MinCachedTweetsGate(identifier, CachedScoredTweets.MinCachedTweetsParam)

)

override val queryTransformer: CandidatePipelineQueryTransformer[

ScoredTweetsQuery,

Seq[t.TimelineQuery]

] = { query =>

val listIds = query.features.map(\_.get(ListIdsFeature)).get

listIds.map { listId =>

t.TimelineQuery(

timelineType = t.TimelineType.List,

timelineId = listId,

maxCount = MaxTweetsToFetchPerList.toShort,

options = Some(t.TimelineQueryOptions(query.clientContext.userId)),

timelineId2 = Some(t.TimelineId(t.TimelineType.List, listId, None))

)

}

}

override def candidateSource: CandidateSource[Seq[t.TimelineQuery], t.Tweet] =

listsCandidateSource

override val featuresFromCandidateSourceTransformers: Seq[

CandidateFeatureTransformer[t.Tweet]

] = Seq(ScoredTweetsListsResponseFeatureTransformer)

override val resultTransformer: CandidatePipelineResultsTransformer[t.Tweet, TweetCandidate] = {

sourceResult => TweetCandidate(id = sourceResult.statusId)

}

override val preFilterFeatureHydrationPhase1: Seq[

BaseCandidateFeatureHydrator[ScoredTweetsQuery, TweetCandidate, \_]

] = Seq(tweetypieStaticEntitiesHydrator)

override val filters: Seq[Filter[ScoredTweetsQuery, TweetCandidate]] =

Seq(ReplyFilter, RetweetFilter)

}