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

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

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

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

import com.twitter.home\_mixer.product.scored\_tweets.filter.RetweetSourceTweetRemovingFilter

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.query\_transformer.TimelineRankerInNetworkQueryTransformer

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

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

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

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

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.PipelineQuery

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

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

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

import javax.inject.Inject

import javax.inject.Singleton

/\*\*

\* Candidate Pipeline Config to fetch in-network tweets from Timeline Ranker's Recycled source

\*/

@Singleton

class ScoredTweetsInNetworkCandidatePipelineConfig @Inject() (

timelineRankerInNetworkCandidateSource: TimelineRankerInNetworkCandidateSource,

replyFeatureHydrator: ReplyFeatureHydrator)

extends CandidatePipelineConfig[

ScoredTweetsQuery,

t.RecapQuery,

t.CandidateTweet,

TweetCandidate

] {

override val identifier: CandidatePipelineIdentifier =

CandidatePipelineIdentifier("ScoredTweetsInNetwork")

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

Some(CandidatePipeline.EnableInNetworkParam)

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

MinCachedTweetsGate(identifier, CachedScoredTweets.MinCachedTweetsParam)

)

override val candidateSource: BaseCandidateSource[t.RecapQuery, t.CandidateTweet] =

timelineRankerInNetworkCandidateSource

override val queryTransformer: CandidatePipelineQueryTransformer[

ScoredTweetsQuery,

t.RecapQuery

] = TimelineRankerInNetworkQueryTransformer(identifier)

override val preFilterFeatureHydrationPhase1: Seq[

BaseCandidateFeatureHydrator[PipelineQuery, TweetCandidate, \_]

] = Seq(RetweetSourceTweetFeatureHydrator)

override def filters: Seq[Filter[ScoredTweetsQuery, TweetCandidate]] = Seq(

RetweetSourceTweetRemovingFilter

)

override val postFilterFeatureHydration: Seq[

BaseCandidateFeatureHydrator[PipelineQuery, TweetCandidate, \_]

] = Seq(IsExtendedReplyFeatureHydrator, replyFeatureHydrator)

override val featuresFromCandidateSourceTransformers: Seq[

CandidateFeatureTransformer[t.CandidateTweet]

] = Seq(ScoredTweetsInNetworkResponseFeatureTransformer)

override val resultTransformer: CandidatePipelineResultsTransformer[

t.CandidateTweet,

TweetCandidate

] = { sourceResult => TweetCandidate(id = sourceResult.tweet.get.id) }

}