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

import com.twitter.home\_mixer.product.scored\_tweets.candidate\_pipeline.CachedScoredTweetsCandidatePipelineConfig.\_

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

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

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

import com.twitter.home\_mixer.{thriftscala => hmt}

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.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 javax.inject.Inject

import javax.inject.Singleton

/\*\*

\* Candidate Pipeline Config that fetches tweets from Scored Tweets Cache.

\*/

@Singleton

class CachedScoredTweetsCandidatePipelineConfig @Inject() (

cachedScoredTweetsCandidateSource: CachedScoredTweetsCandidateSource)

extends CandidatePipelineConfig[

ScoredTweetsQuery,

ScoredTweetsQuery,

hmt.ScoredTweet,

TweetCandidate

] {

override val identifier: CandidatePipelineIdentifier = Identifier

override val queryTransformer: CandidatePipelineQueryTransformer[

ScoredTweetsQuery,

ScoredTweetsQuery

] = identity

override val candidateSource: BaseCandidateSource[ScoredTweetsQuery, hmt.ScoredTweet] =

cachedScoredTweetsCandidateSource

override val featuresFromCandidateSourceTransformers: Seq[

CandidateFeatureTransformer[hmt.ScoredTweet]

] = Seq(CachedScoredTweetsResponseFeatureTransformer)

override val resultTransformer: CandidatePipelineResultsTransformer[

hmt.ScoredTweet,

TweetCandidate

] = { sourceResult => TweetCandidate(id = sourceResult.tweetId) }

}

object CachedScoredTweetsCandidatePipelineConfig {

val Identifier: CandidatePipelineIdentifier = CandidatePipelineIdentifier("CachedScoredTweets")

}