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

import com.twitter.finagle.thrift.ClientId

import com.twitter.home\_mixer.functional\_component.candidate\_source.EarlybirdCandidateSource

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

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.query\_transformer.earlybird.EarlybirdFrsQueryTransformer

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

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.BaseQueryFeatureHydrator

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.search.earlybird.{thriftscala => eb}

import javax.inject.Inject

import javax.inject.Singleton

/\*\*

\* Candidate Pipeline Config that fetches tweets from the earlybird FRS Candidate Source

\*/

@Singleton

class ScoredTweetsEarlybirdFrsCandidatePipelineConfig @Inject() (

earlybirdCandidateSource: EarlybirdCandidateSource,

frsSeedUsersQueryFeatureHydrator: FrsSeedUsersQueryFeatureHydrator,

clientId: ClientId)

extends CandidatePipelineConfig[

ScoredTweetsQuery,

eb.EarlybirdRequest,

eb.ThriftSearchResult,

TweetCandidate

] {

override val identifier: CandidatePipelineIdentifier =

CandidatePipelineIdentifier("ScoredTweetsEarlybirdFrs")

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

MinCachedTweetsGate(identifier, CachedScoredTweets.MinCachedTweetsParam)

)

override val queryFeatureHydration: Seq[

BaseQueryFeatureHydrator[ScoredTweetsQuery, \_]

] = Seq(frsSeedUsersQueryFeatureHydrator)

override val candidateSource: BaseCandidateSource[eb.EarlybirdRequest, eb.ThriftSearchResult] =

earlybirdCandidateSource

override val queryTransformer: CandidatePipelineQueryTransformer[

ScoredTweetsQuery,

eb.EarlybirdRequest

] = EarlybirdFrsQueryTransformer(identifier, clientId = Some(clientId.name))

override val featuresFromCandidateSourceTransformers: Seq[

CandidateFeatureTransformer[eb.ThriftSearchResult]

] = Seq(ScoredTweetsEarlybirdFrsResponseFeatureTransformer)

override val resultTransformer: CandidatePipelineResultsTransformer[

eb.ThriftSearchResult,

TweetCandidate

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

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

}