package com.twitter.home\_mixer.product.following

import com.twitter.home\_mixer.candidate\_pipeline.FollowingEarlybirdResponseFeatureTransformer

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

import com.twitter.home\_mixer.product.following.model.FollowingQuery

import com.twitter.product\_mixer.component\_library.feature\_hydrator.query.social\_graph.SGSFollowedUsersFeature

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.BaseCandidateSource

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 => t}

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class FollowingEarlybirdCandidatePipelineConfig @Inject() (

earlybirdCandidateSource: EarlybirdCandidateSource,

followingEarlybirdQueryTransformer: FollowingEarlybirdQueryTransformer)

extends CandidatePipelineConfig[

FollowingQuery,

t.EarlybirdRequest,

t.ThriftSearchResult,

TweetCandidate

] {

override val identifier: CandidatePipelineIdentifier =

CandidatePipelineIdentifier("FollowingEarlybird")

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

earlybirdCandidateSource

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

NonEmptySeqFeatureGate(SGSFollowedUsersFeature)

)

override val queryTransformer: CandidatePipelineQueryTransformer[

FollowingQuery,

t.EarlybirdRequest

] = followingEarlybirdQueryTransformer

override val featuresFromCandidateSourceTransformers: Seq[

CandidateFeatureTransformer[t.ThriftSearchResult]

] = Seq(FollowingEarlybirdResponseFeatureTransformer)

override val resultTransformer: CandidatePipelineResultsTransformer[

t.ThriftSearchResult,

TweetCandidate

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

}