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

import com.google.inject.Inject

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

import com.twitter.home\_mixer.product.subscribed.model.SubscribedQuery

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

import com.twitter.product\_mixer.component\_library.filter.TweetVisibilityFilter

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.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 => t}

import com.twitter.spam.rtf.thriftscala.SafetyLevel.TimelineHomeSubscribed

import com.twitter.stitch.tweetypie.{TweetyPie => TweetypieStitchClient}

import com.twitter.tweetypie.thriftscala.TweetVisibilityPolicy

class SubscribedEarlybirdCandidatePipelineConfig @Inject() (

earlybirdCandidateSource: EarlybirdCandidateSource,

tweetyPieStitchClient: TweetypieStitchClient,

subscribedEarlybirdQueryTransformer: SubscribedEarlybirdQueryTransformer)

extends CandidatePipelineConfig[

SubscribedQuery,

t.EarlybirdRequest,

t.ThriftSearchResult,

TweetCandidate

] {

override val identifier: CandidatePipelineIdentifier =

CandidatePipelineIdentifier("SubscribedEarlybird")

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

earlybirdCandidateSource

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

NonEmptySeqFeatureGate(SGSSubscribedUsersFeature)

)

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

new TweetVisibilityFilter(

tweetyPieStitchClient,

TweetVisibilityPolicy.UserVisible,

TimelineHomeSubscribed

)

)

override val queryTransformer: CandidatePipelineQueryTransformer[

SubscribedQuery,

t.EarlybirdRequest

] = subscribedEarlybirdQueryTransformer

override val featuresFromCandidateSourceTransformers: Seq[

CandidateFeatureTransformer[t.ThriftSearchResult]

] = Seq(SubscribedEarlybirdResponseFeatureTransformer)

override val resultTransformer: CandidatePipelineResultsTransformer[

t.ThriftSearchResult,

TweetCandidate

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

}