package com.twitter.timelineranker.uteg\_liked\_by\_tweets

import com.twitter.conversions.DurationOps.\_

import com.twitter.timelineranker.config.RequestScopes

import com.twitter.timelineranker.config.RuntimeConfiguration

import com.twitter.timelineranker.parameters.ConfigBuilder

import com.twitter.timelineranker.repository.CandidatesRepositoryBuilder

import com.twitter.timelineranker.visibility.SgsFollowGraphDataFields

import com.twitter.search.earlybird.thriftscala.EarlybirdService

import com.twitter.timelines.util.stats.RequestScope

import com.twitter.util.Duration

class UtegLikedByTweetsRepositoryBuilder(config: RuntimeConfiguration, configBuilder: ConfigBuilder)

extends CandidatesRepositoryBuilder(config) {

override val clientSubId = "uteg\_liked\_by\_tweets"

override val requestScope: RequestScope = RequestScopes.UtegLikedByTweetsSource

override val followGraphDataFieldsToFetch: SgsFollowGraphDataFields.ValueSet =

SgsFollowGraphDataFields.ValueSet(

SgsFollowGraphDataFields.FollowedUserIds,

SgsFollowGraphDataFields.MutuallyFollowingUserIds,

SgsFollowGraphDataFields.MutedUserIds

)

override val searchProcessingTimeout: Duration = 400.milliseconds

override def earlybirdClient(scope: String): EarlybirdService.MethodPerEndpoint =

config.underlyingClients.createEarlybirdClient(

scope = scope,

requestTimeout = 500.milliseconds,

timeout = 900.milliseconds,

retryPolicy = config.underlyingClients.DefaultRetryPolicy

)

def apply(): UtegLikedByTweetsRepository = {

val utegLikedByTweetsSource = new UtegLikedByTweetsSource(

userTweetEntityGraphClient = userTweetEntityGraphClient,

gizmoduckClient = gizmoduckClient,

searchClient = searchClient,

tweetyPieClient = tweetyPieHighQoSClient,

userMetadataClient = userMetadataClient,

followGraphDataProvider = followGraphDataProvider,

contentFeaturesCache = config.underlyingClients.contentFeaturesCache,

statsReceiver = config.statsReceiver

)

new UtegLikedByTweetsRepository(utegLikedByTweetsSource)

}

}