package com.twitter.timelineranker.recap\_hydration

import com.twitter.conversions.DurationOps.\_

import com.twitter.finagle.service.RetryPolicy

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 RecapHydrationRepositoryBuilder(config: RuntimeConfiguration, configBuilder: ConfigBuilder)

extends CandidatesRepositoryBuilder(config) {

override val clientSubId = "feature\_hydration"

override val requestScope: RequestScope = RequestScopes.RecapHydrationSource

override val followGraphDataFieldsToFetch: SgsFollowGraphDataFields.ValueSet =

SgsFollowGraphDataFields.ValueSet(

SgsFollowGraphDataFields.FollowedUserIds,

SgsFollowGraphDataFields.MutuallyFollowingUserIds

)

override val searchProcessingTimeout: Duration = 200.milliseconds //[2]

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

config.underlyingClients.createEarlybirdClient(

scope = scope,

requestTimeout = 500.milliseconds, // [1]

timeout = 500.milliseconds, // [1]

retryPolicy = RetryPolicy.Never

)

def apply(): RecapHydrationRepository = {

val recapHydrationSource = new RecapHydrationSource(

gizmoduckClient,

searchClient,

tweetyPieLowQoSClient,

userMetadataClient,

followGraphDataProvider,

config.underlyingClients.contentFeaturesCache,

config.statsReceiver

)

new RecapHydrationRepository(recapHydrationSource)

}

}