package com.twitter.tweetypie

package backends

import com.twitter.conversions.PercentOps.\_

import com.twitter.conversions.DurationOps.\_

import com.twitter.dataproducts.enrichments.thriftscala.\_

import com.twitter.dataproducts.enrichments.thriftscala.Enricherator

import com.twitter.finagle.thriftmux.MethodBuilder

import com.twitter.servo.util.FutureArrow

object GnipEnricherator {

type HydrateProfileGeo = FutureArrow[ProfileGeoRequest, Seq[ProfileGeoResponse]]

private def methodPerEndpoint(methodBuilder: MethodBuilder) =

Enricherator.MethodPerEndpoint(

methodBuilder

.servicePerEndpoint[Enricherator.ServicePerEndpoint]

.withHydrateProfileGeo(

methodBuilder

.withTimeoutTotal(300.milliseconds)

.withTimeoutPerRequest(100.milliseconds)

.idempotent(maxExtraLoad = 1.percent)

.servicePerEndpoint[Enricherator.ServicePerEndpoint](methodName = "hydrateProfileGeo")

.hydrateProfileGeo

)

)

def fromMethod(methodBuilder: MethodBuilder): GnipEnricherator = {

val mpe = methodPerEndpoint(methodBuilder)

new GnipEnricherator {

override val hydrateProfileGeo: HydrateProfileGeo =

FutureArrow(mpe.hydrateProfileGeo)

}

}

}

trait GnipEnricherator {

import GnipEnricherator.\_

val hydrateProfileGeo: HydrateProfileGeo

}