package com.twitter.cr\_mixer.module.thrift\_client

import com.google.inject.Provides

import com.twitter.app.Flag

import com.twitter.conversions.DurationOps.richDurationFromInt

import com.twitter.cr\_mixer.module.core.TimeoutConfigModule.TweetypieClientTimeoutFlagName

import com.twitter.finagle.ThriftMux

import com.twitter.finagle.mux.ClientDiscardedRequestException

import com.twitter.finagle.service.ReqRep

import com.twitter.finagle.service.ResponseClass

import com.twitter.finagle.service.RetryBudget

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.finatra.mtls.thriftmux.modules.MtlsClient

import com.twitter.inject.Injector

import com.twitter.inject.thrift.modules.ThriftMethodBuilderClientModule

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

import com.twitter.tweetypie.thriftscala.TweetService

import com.twitter.util.Duration

import com.twitter.util.Throw

import javax.inject.Singleton

object TweetyPieClientModule

extends ThriftMethodBuilderClientModule[

TweetService.ServicePerEndpoint,

TweetService.MethodPerEndpoint

]

with MtlsClient {

override val label = "tweetypie"

override val dest = "/s/tweetypie/tweetypie"

private val tweetypieClientTimeout: Flag[Duration] =

flag[Duration](TweetypieClientTimeoutFlagName, "tweetypie client timeout")

override def requestTimeout: Duration = tweetypieClientTimeout()

override def retryBudget: RetryBudget = RetryBudget.Empty

// We bump the success rate from the default of 0.8 to 0.9 since we're dropping the

// consecutive failures part of the default policy.

override def configureThriftMuxClient(

injector: Injector,

client: ThriftMux.Client

): ThriftMux.Client =

super

.configureThriftMuxClient(injector, client)

.withStatsReceiver(injector.instance[StatsReceiver].scope("clnt"))

.withSessionQualifier

.successRateFailureAccrual(successRate = 0.9, window = 30.seconds)

.withResponseClassifier {

case ReqRep(\_, Throw(\_: ClientDiscardedRequestException)) => ResponseClass.Ignorable

}

@Provides

@Singleton

def providesTweetyPie(

tweetyPieService: TweetService.MethodPerEndpoint

): STweetyPie = {

STweetyPie(tweetyPieService)

}

}