package com.twitter.tsp

package modules

import com.google.inject.Module

import com.google.inject.Provides

import com.twitter.conversions.DurationOps.richDurationFromInt

import com.twitter.finagle.ThriftMux

import com.twitter.finagle.mtls.authentication.ServiceIdentifier

import com.twitter.finagle.mtls.client.MtlsStackClient.MtlsThriftMuxClientSyntax

import com.twitter.finagle.mux.ClientDiscardedRequestException

import com.twitter.finagle.service.ReqRep

import com.twitter.finagle.service.ResponseClass

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.finagle.thrift.ClientId

import com.twitter.inject.Injector

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

import com.twitter.tweetypie.thriftscala.TweetService

import com.twitter.util.Duration

import com.twitter.util.Throw

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

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

import javax.inject.Singleton

object TweetyPieClientModule

extends ThriftMethodBuilderClientModule[

TweetService.ServicePerEndpoint,

TweetService.MethodPerEndpoint

]

with MtlsClient {

override val label = "tweetypie"

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

override val requestTimeout: Duration = 450.milliseconds

override val modules: Seq[Module] = Seq(TSPClientIdModule)

// 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)

.withMutualTls(injector.instance[ServiceIdentifier])

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

.withClientId(injector.instance[ClientId])

.withResponseClassifier {

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

}

.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)

}

}