package com.twitter.home\_mixer.module

import com.google.inject.Provides

import com.twitter.conversions.DurationOps.\_

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

import com.twitter.finagle.thrift.ClientId

import com.twitter.finagle.thriftmux.MethodBuilder

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

import com.twitter.inject.Injector

import com.twitter.inject.annotations.Flags

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

import com.twitter.stitch.tweetypie.TweetyPie

import com.twitter.tweetypie.thriftscala.TweetService

import com.twitter.util.Duration

import javax.inject.Singleton

/\*\*

\* Idempotent Tweetypie Thrift and Stitch client.

\*/

object TweetypieClientModule

extends ThriftMethodBuilderClientModule[

TweetService.ServicePerEndpoint,

TweetService.MethodPerEndpoint

]

with MtlsClient {

private val TimeoutRequest = "tweetypie.timeout\_request"

private val TimeoutTotal = "tweetypie.timeout\_total"

flag[Duration](TimeoutRequest, 1000.millis, "Timeout per request")

flag[Duration](TimeoutTotal, 1000.millis, "Total timeout")

override val label: String = "tweetypie"

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

@Singleton

@Provides

def providesTweetypieStitchClient(tweetService: TweetService.MethodPerEndpoint): TweetyPie =

new TweetyPie(tweetService)

/\*\*

\* TweetyPie client id must be in the form of {service.env} or it will not be treated as an

\* unauthorized client

\*/

override protected def clientId(injector: Injector): ClientId = {

val serviceIdentifier = injector.instance[ServiceIdentifier]

ClientId(s"${serviceIdentifier.service}.${serviceIdentifier.environment}")

}

override protected def configureMethodBuilder(

injector: Injector,

methodBuilder: MethodBuilder

): MethodBuilder = {

val timeoutRequest = injector.instance[Duration](Flags.named(TimeoutRequest))

val timeoutTotal = injector.instance[Duration](Flags.named(TimeoutTotal))

methodBuilder

.withTimeoutPerRequest(timeoutRequest)

.withTimeoutTotal(timeoutTotal)

}

override protected def sessionAcquisitionTimeout: Duration = 500.millis

}