package com.twitter.product\_mixer.component\_library.module

import com.google.inject.Provides

import com.twitter.conversions.DurationOps.\_

import com.twitter.conversions.PercentOps.\_

import com.twitter.finagle.thriftmux.MethodBuilder

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

import com.twitter.gizmoduck.thriftscala.UserService

import com.twitter.inject.Injector

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

import com.twitter.stitch.gizmoduck.Gizmoduck

import com.twitter.util.Duration

import javax.inject.Singleton

/\*\*

\* Implementation with reasonable defaults for an idempotent Gizmoduck Thrift and Stitch client.

\*

\* Note that the per request and total timeouts configured in this module are meant to represent a

\* reasonable starting point only. These were selected based on common practice, and should not be

\* assumed to be optimal for any particular use case. If you are interested in further tuning the

\* settings in this module, it is recommended to create local copy for your service.

\*/

object GizmoduckClientModule

extends ThriftMethodBuilderClientModule[

UserService.ServicePerEndpoint,

UserService.MethodPerEndpoint

]

with MtlsClient {

override val label: String = "gizmoduck"

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

@Singleton

@Provides

def provideGizmoduckStitchClient(userService: UserService.MethodPerEndpoint): Gizmoduck =

new Gizmoduck(userService)

override protected def configureMethodBuilder(

injector: Injector,

methodBuilder: MethodBuilder

): MethodBuilder =

methodBuilder

.withTimeoutPerRequest(200.milliseconds)

.withTimeoutTotal(400.milliseconds)

.idempotent(1.percent)

override protected def sessionAcquisitionTimeout: Duration = 500.milliseconds

}