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.inject.Injector

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

import com.twitter.stitch.timelineservice.TimelineService

import com.twitter.timelineservice.{thriftscala => t}

import com.twitter.util.Duration

import javax.inject.Singleton

object TimelineServiceClientModule

extends ThriftMethodBuilderClientModule[

t.TimelineService.ServicePerEndpoint,

t.TimelineService.MethodPerEndpoint

]

with MtlsClient {

override val label = "timelineservice"

override val dest = "/s/timelineservice/timelineservice"

override protected def configureMethodBuilder(

injector: Injector,

methodBuilder: MethodBuilder

): MethodBuilder = {

methodBuilder

.withTimeoutPerRequest(1200.millis)

.withTimeoutTotal(2400.millis)

.idempotent(1.percent)

}

override protected def sessionAcquisitionTimeout: Duration = 500.milliseconds

@Singleton

@Provides

def providesTimelineServiceStitchClient(

client: t.TimelineService.MethodPerEndpoint

): TimelineService = {

new TimelineService(client)

}

}