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

import com.twitter.conversions.DurationOps.\_

import com.twitter.finagle.ThriftMux

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

import com.twitter.finagle.mtls.client.MtlsStackClient.\_

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.timelineranker.{thriftscala => t}

import com.twitter.util.Duration

import org.apache.thrift.protocol.TCompactProtocol

object TimelineRankerClientModule

extends ThriftMethodBuilderClientModule[

t.TimelineRanker.ServicePerEndpoint,

t.TimelineRanker.MethodPerEndpoint

]

with MtlsClient {

override val label = "timeline-ranker"

override val dest = "/s/timelineranker/timelineranker:compactthrift"

override protected def configureMethodBuilder(

injector: Injector,

methodBuilder: MethodBuilder

): MethodBuilder = {

methodBuilder

.withTimeoutPerRequest(750.millis)

.withTimeoutTotal(750.millis)

}

override def configureThriftMuxClient(

injector: Injector,

client: ThriftMux.Client

): ThriftMux.Client = {

val serviceIdentifier = injector.instance[ServiceIdentifier]

super

.configureThriftMuxClient(injector, client)

.withProtocolFactory(new TCompactProtocol.Factory())

.withMutualTls(serviceIdentifier)

.withPerEndpointStats

}

override protected def sessionAcquisitionTimeout: Duration = 500.milliseconds

}