package com.twitter.cr\_mixer.module.thrift\_client

import com.twitter.app.Flag

import com.twitter.cr\_mixer.module.core.TimeoutConfigModule.UtegClientTimeoutFlagName

import com.twitter.finagle.ThriftMux

import com.twitter.finagle.mux.ClientDiscardedRequestException

import com.twitter.finagle.service.ReqRep

import com.twitter.finagle.service.ResponseClass

import com.twitter.finagle.service.RetryBudget

import com.twitter.finagle.stats.StatsReceiver

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

import com.twitter.inject.Injector

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

import com.twitter.recos.user\_tweet\_entity\_graph.thriftscala.UserTweetEntityGraph

import com.twitter.util.Duration

import com.twitter.util.Throw

object UserTweetEntityGraphClientModule

extends ThriftMethodBuilderClientModule[

UserTweetEntityGraph.ServicePerEndpoint,

UserTweetEntityGraph.MethodPerEndpoint

]

with MtlsClient {

override val label = "user-tweet-entity-graph"

override val dest = "/s/cassowary/user\_tweet\_entity\_graph"

private val userTweetEntityGraphClientTimeout: Flag[Duration] =

flag[Duration](UtegClientTimeoutFlagName, "user tweet entity graph client timeout")

override def requestTimeout: Duration = userTweetEntityGraphClientTimeout()

override def retryBudget: RetryBudget = RetryBudget.Empty

override def configureThriftMuxClient(

injector: Injector,

client: ThriftMux.Client

): ThriftMux.Client =

super

.configureThriftMuxClient(injector, client)

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

.withResponseClassifier {

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

}

}