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.service.Retries

import com.twitter.finagle.service.RetryPolicy

import com.twitter.finagle.ssl.OpportunisticTls

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.home\_mixer.param.HomeMixerInjectionNames.BatchedStratoClientWithModerateTimeout

import com.twitter.inject.TwitterModule

import com.twitter.strato.client.Client

import com.twitter.strato.client.Strato

import com.twitter.util.Try

import javax.inject.Named

import javax.inject.Singleton

object OptimizedStratoClientModule extends TwitterModule {

private val ModerateStratoServerClientRequestTimeout = 500.millis

private val DefaultRetryPartialFunction: PartialFunction[Try[Nothing], Boolean] =

RetryPolicy.TimeoutAndWriteExceptionsOnly

.orElse(RetryPolicy.ChannelClosedExceptionsOnly)

protected def mkRetryPolicy(tries: Int): RetryPolicy[Try[Nothing]] =

RetryPolicy.tries(tries, DefaultRetryPartialFunction)

@Singleton

@Provides

@Named(BatchedStratoClientWithModerateTimeout)

def providesStratoClient(

serviceIdentifier: ServiceIdentifier,

statsReceiver: StatsReceiver

): Client = {

Strato.client

.withMutualTls(serviceIdentifier, opportunisticLevel = OpportunisticTls.Required)

.withSession.acquisitionTimeout(500.milliseconds)

.withRequestTimeout(ModerateStratoServerClientRequestTimeout)

.withPerRequestTimeout(ModerateStratoServerClientRequestTimeout)

.withRpcBatchSize(5)

.configured(Retries.Policy(mkRetryPolicy(1)))

.withStatsReceiver(statsReceiver.scope("strato\_client"))

.build()

}

}