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

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.annotations.Flags

import com.twitter.inject.Injector

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

import com.twitter.account\_recommendations\_mixer.thriftscala.AccountRecommendationsMixer

import com.twitter.util.Duration

/\*\*

\* Implementation with reasonable defaults for an idempotent Account Recommendations Mixer Thrift 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 AccountRecommendationsMixerModule

extends ThriftMethodBuilderClientModule[

AccountRecommendationsMixer.ServicePerEndpoint,

AccountRecommendationsMixer.MethodPerEndpoint

]

with MtlsClient {

final val AccountRecommendationsMixerTimeoutPerRequest =

"account\_recommendations\_mixer.timeout\_per\_request"

final val AccountRecommendationsMixerTimeoutTotal = "account\_recommendations\_mixer.timeout\_total"

flag[Duration](

name = AccountRecommendationsMixerTimeoutPerRequest,

default = 800.milliseconds,

help = "Timeout per request for AccountRecommendationsMixer")

flag[Duration](

name = AccountRecommendationsMixerTimeoutTotal,

default = 1200.milliseconds,

help = "Timeout total for AccountRecommendationsMixer")

override val label: String = "account-recs-mixer"

override val dest: String = "/s/account-recs-mixer/account-recs-mixer:thrift"

override protected def configureMethodBuilder(

injector: Injector,

methodBuilder: MethodBuilder

): MethodBuilder = {

val timeOutPerRequest: Duration = injector

.instance[Duration](Flags.named(AccountRecommendationsMixerTimeoutPerRequest))

val timeOutTotal: Duration =

injector.instance[Duration](Flags.named(AccountRecommendationsMixerTimeoutTotal))

methodBuilder

.withTimeoutPerRequest(timeOutPerRequest)

.withTimeoutTotal(timeOutTotal)

.idempotent(5.percent)

}

override protected def sessionAcquisitionTimeout: Duration = 500.milliseconds

}