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

import com.google.inject.Provides

import com.twitter.conversions.PercentOps.\_

import com.twitter.cr\_mixer.model.ModuleNames

import com.twitter.cr\_mixer.config.TimeoutConfig

import com.twitter.finagle.ThriftMux

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

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

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.finagle.thrift.ClientId

import com.twitter.inject.TwitterModule

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

import javax.inject.Named

import javax.inject.Singleton

object SimClustersAnnServiceClientModule extends TwitterModule {

@Provides

@Singleton

@Named(ModuleNames.ProdSimClustersANNServiceClientName)

def providesProdSimClustersANNServiceClient(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

): t.SimClustersANNService.MethodPerEndpoint = {

val label = "simclusters-ann-server"

val dest = "/s/simclusters-ann/simclusters-ann"

buildClient(serviceIdentifier, clientId, timeoutConfig, statsReceiver, dest, label)

}

@Provides

@Singleton

@Named(ModuleNames.ExperimentalSimClustersANNServiceClientName)

def providesExperimentalSimClustersANNServiceClient(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

): t.SimClustersANNService.MethodPerEndpoint = {

val label = "simclusters-ann-experimental-server"

val dest = "/s/simclusters-ann/simclusters-ann-experimental"

buildClient(serviceIdentifier, clientId, timeoutConfig, statsReceiver, dest, label)

}

@Provides

@Singleton

@Named(ModuleNames.SimClustersANNServiceClientName1)

def providesSimClustersANNServiceClient1(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

): t.SimClustersANNService.MethodPerEndpoint = {

val label = "simclusters-ann-server-1"

val dest = "/s/simclusters-ann/simclusters-ann-1"

buildClient(serviceIdentifier, clientId, timeoutConfig, statsReceiver, dest, label)

}

@Provides

@Singleton

@Named(ModuleNames.SimClustersANNServiceClientName2)

def providesSimClustersANNServiceClient2(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

): t.SimClustersANNService.MethodPerEndpoint = {

val label = "simclusters-ann-server-2"

val dest = "/s/simclusters-ann/simclusters-ann-2"

buildClient(serviceIdentifier, clientId, timeoutConfig, statsReceiver, dest, label)

}

@Provides

@Singleton

@Named(ModuleNames.SimClustersANNServiceClientName3)

def providesSimClustersANNServiceClient3(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

): t.SimClustersANNService.MethodPerEndpoint = {

val label = "simclusters-ann-server-3"

val dest = "/s/simclusters-ann/simclusters-ann-3"

buildClient(serviceIdentifier, clientId, timeoutConfig, statsReceiver, dest, label)

}

@Provides

@Singleton

@Named(ModuleNames.SimClustersANNServiceClientName5)

def providesSimClustersANNServiceClient5(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

): t.SimClustersANNService.MethodPerEndpoint = {

val label = "simclusters-ann-server-5"

val dest = "/s/simclusters-ann/simclusters-ann-5"

buildClient(serviceIdentifier, clientId, timeoutConfig, statsReceiver, dest, label)

}

@Provides

@Singleton

@Named(ModuleNames.SimClustersANNServiceClientName4)

def providesSimClustersANNServiceClient4(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

): t.SimClustersANNService.MethodPerEndpoint = {

val label = "simclusters-ann-server-4"

val dest = "/s/simclusters-ann/simclusters-ann-4"

buildClient(serviceIdentifier, clientId, timeoutConfig, statsReceiver, dest, label)

}

private def buildClient(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

dest: String,

label: String

): t.SimClustersANNService.MethodPerEndpoint = {

val stats = statsReceiver.scope("clnt")

val thriftClient = ThriftMux.client

.withMutualTls(serviceIdentifier)

.withClientId(clientId)

.withLabel(label)

.withStatsReceiver(stats)

.methodBuilder(dest)

.idempotent(5.percent)

.withTimeoutPerRequest(timeoutConfig.annServiceClientTimeout)

.withRetryDisabled

.servicePerEndpoint[t.SimClustersANNService.ServicePerEndpoint]

ThriftMux.Client.methodPerEndpoint(thriftClient)

}

}