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

import com.google.inject.Provides

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

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

import com.twitter.finagle.Http

import com.twitter.finagle.grpc.FinagleChannelBuilder

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

import com.twitter.finagle.mtls.client.MtlsStackClient.MtlsStackClientSyntax

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import com.twitter.util.Duration

import io.grpc.ManagedChannel

import javax.inject.Named

import javax.inject.Singleton

object NaviGRPCClientModule extends TwitterModule {

val maxRetryAttempts = 3

@Provides

@Singleton

@Named(ModuleNames.HomeNaviGRPCClient)

def providesHomeNaviGRPCClient(

serviceIdentifier: ServiceIdentifier,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

): ManagedChannel = {

val label = "navi-wals-recommended-tweets-home-client"

val dest = "/s/ads-prediction/navi-wals-recommended-tweets-home"

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

}

@Provides

@Singleton

@Named(ModuleNames.AdsFavedNaviGRPCClient)

def providesAdsFavedNaviGRPCClient(

serviceIdentifier: ServiceIdentifier,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

): ManagedChannel = {

val label = "navi-wals-ads-faved-tweets"

val dest = "/s/ads-prediction/navi-wals-ads-faved-tweets"

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

}

@Provides

@Singleton

@Named(ModuleNames.AdsMonetizableNaviGRPCClient)

def providesAdsMonetizableNaviGRPCClient(

serviceIdentifier: ServiceIdentifier,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

): ManagedChannel = {

val label = "navi-wals-ads-monetizable-tweets"

val dest = "/s/ads-prediction/navi-wals-ads-monetizable-tweets"

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

}

private def buildClient(

serviceIdentifier: ServiceIdentifier,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

dest: String,

label: String

): ManagedChannel = {

val stats = statsReceiver.scope("clnt").scope(label)

val client = Http.client

.withLabel(label)

.withMutualTls(serviceIdentifier)

.withRequestTimeout(timeoutConfig.naviRequestTimeout)

.withTransport.connectTimeout(Duration.fromMilliseconds(10000))

.withSession.acquisitionTimeout(Duration.fromMilliseconds(20000))

.withStatsReceiver(stats)

.withHttpStats

FinagleChannelBuilder

.forTarget(dest)

.overrideAuthority("rustserving")

.maxRetryAttempts(maxRetryAttempts)

.enableRetryForStatus(io.grpc.Status.RESOURCE\_EXHAUSTED)

.enableRetryForStatus(io.grpc.Status.UNKNOWN)

.enableUnsafeFullyBufferingMode()

.httpClient(client)

.build()

}

}