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

import com.google.inject.Provides

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

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.finatra.httpclient.HttpClient

import com.twitter.inject.TwitterModule

import com.twitter.inject.annotations.Flag

import com.twitter.product\_mixer.component\_library.module.http.FinagleHttpClientModule.HttpClientAcquisitionTimeout

import com.twitter.product\_mixer.component\_library.module.http.FinagleHttpClientModule.HttpClientConnectTimeout

import com.twitter.product\_mixer.component\_library.module.http.FinagleHttpClientModule.HttpClientRequestTimeout

import com.twitter.product\_mixer.shared\_library.http\_client.FinagleHttpClientBuilder.buildFinagleHttpClientMutualTls

import com.twitter.product\_mixer.shared\_library.http\_client.HttpHostPort

import com.twitter.util.Duration

import com.twitter.util.jackson.ScalaObjectMapper

import javax.inject.Named

import javax.inject.Singleton

object FinatraHttpClientModule extends TwitterModule {

final val HttpClientHost = "http\_client.host"

final val HttpClientPort = "http\_client.port"

flag[String](HttpClientHost, "Host that the client will connect to")

flag[Int](HttpClientPort, 443, "Port that the client will connect to")

final val FinatraHttpClient = "FinatraHttpClient"

/\*\*

\* Build a Finatra HTTP client for a host. The Finatra HTTP client can be helpful (as opposed to

\* the base Finagle HTTP Client), as it provides built-in JSON response parsing and other

\* convenience methods

\*

\* Note that the timeouts configured in this module are meant to be a reasonable starting point

\* only. To further tuning the settings, either override the flags or create local copy of the module.

\*

\* @param requestTimeout HTTP client request timeout

\* @param connectTimeout HTTP client transport connect timeout

\* @param acquisitionTimeout HTTP client session acquisition timeout

\* @param host Host to build Finatra client

\* @param port Port to build Finatra client

\* @param scalaObjectMapper Object mapper used by the built-in JSON response parsing

\* @param serviceIdentifier Service ID used to S2S Auth

\* @param statsReceiver Stats

\*

\* @return Finatra HTTP client

\*/

@Provides

@Singleton

@Named(FinatraHttpClient)

def providesFinatraHttpClient(

@Flag(HttpClientRequestTimeout) requestTimeout: Duration,

@Flag(HttpClientConnectTimeout) connectTimeout: Duration,

@Flag(HttpClientAcquisitionTimeout) acquisitionTimeout: Duration,

@Flag(HttpClientHost) host: String,

@Flag(HttpClientPort) port: Int,

scalaObjectMapper: ScalaObjectMapper,

serviceIdentifier: ServiceIdentifier,

statsReceiver: StatsReceiver

): HttpClient = {

val finagleHttpClient = buildFinagleHttpClientMutualTls(

requestTimeout = requestTimeout,

connectTimeout = connectTimeout,

acquisitionTimeout = acquisitionTimeout,

serviceIdentifier = serviceIdentifier,

statsReceiver = statsReceiver

)

val hostPort = HttpHostPort(host, port)

val finagleHttpService = finagleHttpClient.newService(hostPort.toString)

new HttpClient(

hostname = hostPort.host,

httpService = finagleHttpService,

mapper = scalaObjectMapper

)

}

}