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

import com.google.inject.Provides

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.component\_library.module.http.FinagleHttpClientWithProxyModule.HttpClientWithProxyRemoteHost

import com.twitter.product\_mixer.component\_library.module.http.FinagleHttpClientWithProxyModule.HttpClientWithProxyRemotePort

import com.twitter.product\_mixer.component\_library.module.http.FinagleHttpClientWithProxyModule.HttpClientWithProxyTwitterHost

import com.twitter.product\_mixer.component\_library.module.http.FinagleHttpClientWithProxyModule.HttpClientWithProxyTwitterPort

import com.twitter.product\_mixer.core.module.product\_mixer\_flags.ProductMixerFlagModule.ServiceLocal

import com.twitter.product\_mixer.shared\_library.http\_client.FinagleHttpClientWithProxyBuilder.buildFinagleHttpClientWithProxy

import com.twitter.product\_mixer.shared\_library.http\_client.FinagleHttpClientWithProxyBuilder.buildFinagleHttpServiceWithProxy

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 FinatraHttpClientWithProxyModule extends TwitterModule {

final val FinatraHttpClientWithProxy = "FinagleHttpClientWithProxy"

/\*\*

\* Build a Finatra HTTP client with Egress Proxy support 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 proxyTwitterHost Twitter egress proxy host

\* @param proxyTwitterPort Twitter egress proxy port

\* @param proxyRemoteHost Remote proxy host

\* @param proxyRemotePort Remote proxy port

\* @param requestTimeout HTTP client request timeout

\* @param connectTimeout HTTP client transport connect timeout

\* @param acquisitionTimeout HTTP client session acquisition timeout

\* @param isServiceLocal Local deployment for testing

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

\* @param statsReceiver Stats

\*

\* @return Finatra HTTP client with Egress Proxy support for a host

\*/

@Provides

@Singleton

@Named(FinatraHttpClientWithProxy)

def providesFinatraHttpClientWithProxy(

@Flag(HttpClientWithProxyTwitterHost) proxyTwitterHost: String,

@Flag(HttpClientWithProxyTwitterPort) proxyTwitterPort: Int,

@Flag(HttpClientWithProxyRemoteHost) proxyRemoteHost: String,

@Flag(HttpClientWithProxyRemotePort) proxyRemotePort: Int,

@Flag(HttpClientRequestTimeout) requestTimeout: Duration,

@Flag(HttpClientConnectTimeout) connectTimeout: Duration,

@Flag(HttpClientAcquisitionTimeout) acquisitionTimeout: Duration,

@Flag(ServiceLocal) isServiceLocal: Boolean,

scalaObjectMapper: ScalaObjectMapper,

statsReceiver: StatsReceiver

): HttpClient = {

val twitterProxyHostPort = HttpHostPort(proxyTwitterHost, proxyTwitterPort)

val proxyRemoteHostPort = HttpHostPort(proxyRemoteHost, proxyRemotePort)

val finagleHttpClientWithProxy =

buildFinagleHttpClientWithProxy(

twitterProxyHostPort = twitterProxyHostPort,

remoteProxyHostPort = proxyRemoteHostPort,

requestTimeout = requestTimeout,

connectTimeout = connectTimeout,

acquisitionTimeout = acquisitionTimeout,

statsReceiver = statsReceiver

)

val finagleHttpServiceWithProxy =

buildFinagleHttpServiceWithProxy(

finagleHttpClientWithProxy = finagleHttpClientWithProxy,

twitterProxyHostPort = twitterProxyHostPort

)

new HttpClient(

hostname = twitterProxyHostPort.host,

httpService = finagleHttpServiceWithProxy,

mapper = scalaObjectMapper

)

}

}