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

import com.google.inject.Provides

import com.twitter.finagle.Http

import com.twitter.finagle.stats.StatsReceiver

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.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.HttpHostPort

import com.twitter.util.Duration

import javax.inject.Named

import javax.inject.Singleton

object FinagleHttpClientWithProxyModule extends TwitterModule {

final val HttpClientWithProxyTwitterHost = "http\_client.proxy.twitter\_host"

final val HttpClientWithProxyTwitterPort = "http\_client.proxy.twitter\_port"

final val HttpClientWithProxyRemoteHost = "http\_client.proxy.remote\_host"

final val HttpClientWithProxyRemotePort = "http\_client.proxy.remote\_port"

flag[String](

HttpClientWithProxyTwitterHost,

"httpproxy.local.twitter.com",

"Twitter egress proxy host")

flag[Int](HttpClientWithProxyTwitterPort, 3128, "Twitter egress proxy port")

flag[String](HttpClientWithProxyRemoteHost, "Host that the proxy will connect to")

flag[Int](HttpClientWithProxyRemotePort, 443, "Port that the proxy will connect to")

final val FinagleHttpClientWithProxy = "FinagleHttpClientWithProxy"

/\*\*

\* Provide a Finagle HTTP client with Egress Proxy support

\*

\* 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 If this is a Local deployment for testing

\* @param statsReceiver Stats

\*

\* @return Finagle HTTP client with Egress Proxy support

\*/

@Provides

@Singleton

@Named(FinagleHttpClientWithProxy)

def providesFinagleHttpClientWithProxy(

@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,

statsReceiver: StatsReceiver

): Http.Client = {

val twitterProxyHostPort = HttpHostPort(proxyTwitterHost, proxyTwitterPort)

val remoteProxyHostPort = HttpHostPort(proxyRemoteHost, proxyRemotePort)

buildFinagleHttpClientWithProxy(

twitterProxyHostPort = twitterProxyHostPort,

remoteProxyHostPort = remoteProxyHostPort,

requestTimeout = requestTimeout,

connectTimeout = connectTimeout,

acquisitionTimeout = acquisitionTimeout,

statsReceiver = statsReceiver

)

}

}