package com.twitter.product\_mixer.core.module

import com.google.inject.Provides

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

import com.twitter.finagle.ssl.OpportunisticTls

import com.twitter.inject.TwitterModule

import com.twitter.inject.annotations.Flag

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

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

import com.twitter.strato.client.Client

import com.twitter.strato.client.Strato

import com.twitter.util.Duration

import javax.inject.Singleton

/\*\*

\* Product Mixer prefers to use a single strato client module over having a variety with different

\* timeouts. Latency Budgets in Product Mixer systems should be defined at the application layer.

\*/

object StratoClientModule extends TwitterModule {

@Provides

@Singleton

def providesStratoClient(

serviceIdentifier: ServiceIdentifier,

@Flag(ServiceLocal) isServiceLocal: Boolean,

@Flag(StratoLocalRequestTimeout) timeout: Option[Duration]

): Client = {

val stratoClient = Strato.client.withMutualTls(serviceIdentifier, OpportunisticTls.Required)

// For local development it can be useful to have a larger timeout than the Strato default of

// 280ms. We strongly discourage setting client-level timeouts outside of this use-case. We

// recommend setting an overall timeout for your pipeline's end-to-end running time.

if (isServiceLocal && timeout.isDefined)

stratoClient.withRequestTimeout(timeout.get).build()

else {

stratoClient.build()

}

}

}