package com.twitter.follow\_recommendations.common.clients.deepbirdv2

import com.google.inject.Provides

import com.google.inject.name.Named

import com.twitter.bijection.scrooge.TBinaryProtocol

import com.twitter.conversions.DurationOps.\_

import com.twitter.cortex.deepbird.thriftjava.DeepbirdPredictionService

import com.twitter.finagle.ThriftMux

import com.twitter.finagle.builder.ClientBuilder

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

import com.twitter.finagle.mtls.client.MtlsStackClient.\_

import com.twitter.finagle.stats.NullStatsReceiver

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.finagle.thrift.ClientId

import com.twitter.finagle.thrift.RichClientParam

import com.twitter.follow\_recommendations.common.constants.GuiceNamedConstants

import com.twitter.inject.TwitterModule

/\*\*

\* Module that provides multiple deepbirdv2 prediction service clients

\* We use the java api since data records are native java objects and we want to reduce overhead

\* while serializing/deserializing data.

\*/

object DeepBirdV2PredictionServiceClientModule extends TwitterModule {

val RequestTimeout = 300.millis

private def getDeepbirdPredictionServiceClient(

clientId: ClientId,

label: String,

dest: String,

statsReceiver: StatsReceiver,

serviceIdentifier: ServiceIdentifier

): DeepbirdPredictionService.ServiceToClient = {

val clientStatsReceiver = statsReceiver.scope("clnt")

val mTlsClient = ThriftMux.client.withClientId(clientId).withMutualTls(serviceIdentifier)

new DeepbirdPredictionService.ServiceToClient(

ClientBuilder()

.name(label)

.stack(mTlsClient)

.dest(dest)

.requestTimeout(RequestTimeout)

.reportHostStats(NullStatsReceiver)

.build(),

RichClientParam(

new TBinaryProtocol.Factory(),

clientStats = clientStatsReceiver

)

)

}

@Provides

@Named(GuiceNamedConstants.WTF\_PROD\_DEEPBIRDV2\_CLIENT)

def providesWtfProdDeepbirdV2PredictionService(

clientId: ClientId,

statsReceiver: StatsReceiver,

serviceIdentifier: ServiceIdentifier

): DeepbirdPredictionService.ServiceToClient = {

getDeepbirdPredictionServiceClient(

clientId = clientId,

label = "WtfProdDeepbirdV2PredictionService",

dest = "/s/cassowary/deepbirdv2-hermit-wtf",

statsReceiver = statsReceiver,

serviceIdentifier = serviceIdentifier

)

}

}