package com.twitter.cr\_mixer.module.thrift\_client

import com.google.inject.Provides

import com.twitter.ann.common.thriftscala.AnnQueryService

import com.twitter.conversions.DurationOps.\_

import com.twitter.conversions.PercentOps.\_

import com.twitter.cr\_mixer.config.TimeoutConfig

import com.twitter.finagle.ThriftMux

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

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

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.finagle.thrift.ClientId

import com.twitter.inject.TwitterModule

import javax.inject.Named

import javax.inject.Singleton

object AnnQueryServiceClientModule extends TwitterModule {

final val DebuggerDemoAnnServiceClientName = "DebuggerDemoAnnServiceClient"

@Provides

@Singleton

@Named(DebuggerDemoAnnServiceClientName)

def debuggerDemoAnnServiceClient(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

statsReceiver: StatsReceiver,

timeoutConfig: TimeoutConfig,

): AnnQueryService.MethodPerEndpoint = {

// This ANN is built from the embeddings in src/scala/com/twitter/wtf/beam/bq\_embedding\_export/sql/MlfExperimentalTweetEmbeddingScalaDataset.sql

// Change the above sql if you want to build the index from a diff embedding

val dest = "/s/cassowary/mlf-experimental-ann-service"

val label = "experimental-ann"

buildClient(serviceIdentifier, clientId, timeoutConfig, statsReceiver, dest, label)

}

final val TwHINUuaAnnServiceClientName = "TwHINUuaAnnServiceClient"

@Provides

@Singleton

@Named(TwHINUuaAnnServiceClientName)

def twhinUuaAnnServiceClient(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

statsReceiver: StatsReceiver,

timeoutConfig: TimeoutConfig,

): AnnQueryService.MethodPerEndpoint = {

val dest = "/s/cassowary/twhin-uua-ann-service"

val label = "twhin\_uua\_ann"

buildClient(serviceIdentifier, clientId, timeoutConfig, statsReceiver, dest, label)

}

final val TwHINRegularUpdateAnnServiceClientName = "TwHINRegularUpdateAnnServiceClient"

@Provides

@Singleton

@Named(TwHINRegularUpdateAnnServiceClientName)

def twHINRegularUpdateAnnServiceClient(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

statsReceiver: StatsReceiver,

timeoutConfig: TimeoutConfig,

): AnnQueryService.MethodPerEndpoint = {

val dest = "/s/cassowary/twhin-regular-update-ann-service"

val label = "twhin\_regular\_update"

buildClient(serviceIdentifier, clientId, timeoutConfig, statsReceiver, dest, label)

}

final val TwoTowerFavAnnServiceClientName = "TwoTowerFavAnnServiceClient"

@Provides

@Singleton

@Named(TwoTowerFavAnnServiceClientName)

def twoTowerFavAnnServiceClient(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

statsReceiver: StatsReceiver,

timeoutConfig: TimeoutConfig,

): AnnQueryService.MethodPerEndpoint = {

val dest = "/s/cassowary/tweet-rec-two-tower-fav-ann"

val label = "tweet\_rec\_two\_tower\_fav\_ann"

buildClient(serviceIdentifier, clientId, timeoutConfig, statsReceiver, dest, label)

}

private def buildClient(

serviceIdentifier: ServiceIdentifier,

clientId: ClientId,

timeoutConfig: TimeoutConfig,

statsReceiver: StatsReceiver,

dest: String,

label: String

): AnnQueryService.MethodPerEndpoint = {

val thriftClient = ThriftMux.client

.withMutualTls(serviceIdentifier)

.withClientId(clientId)

.withLabel(label)

.withStatsReceiver(statsReceiver)

.withTransport.connectTimeout(500.milliseconds)

.withSession.acquisitionTimeout(500.milliseconds)

.methodBuilder(dest)

.withTimeoutPerRequest(timeoutConfig.annServiceClientTimeout)

.withRetryDisabled

.idempotent(5.percent)

.servicePerEndpoint[AnnQueryService.ServicePerEndpoint]

ThriftMux.Client.methodPerEndpoint(thriftClient)

}

}