package com.twitter.home\_mixer.module

import com.google.inject.Provides

import com.twitter.conversions.DurationOps.\_

import com.twitter.finagle.Http

import com.twitter.finagle.grpc.FinagleChannelBuilder

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

import com.twitter.finagle.mtls.client.MtlsStackClient.MtlsStackClientSyntax

import com.twitter.inject.TwitterModule

import com.twitter.timelines.clients.predictionservice.PredictionGRPCService

import com.twitter.util.Duration

import io.grpc.ManagedChannel

import javax.inject.Singleton

object NaviModelClientModule extends TwitterModule {

@Singleton

@Provides

def providesPredictionGRPCService(

serviceIdentifier: ServiceIdentifier,

): PredictionGRPCService = {

// Wily path to the ML Model service (e.g. /s/ml-serving/navi-explore-ranker).

val modelPath = "/s/ml-serving/navi\_home\_recap\_onnx"

val MaxPredictionTimeoutMs: Duration = 500.millis

val ConnectTimeoutMs: Duration = 200.millis

val AcquisitionTimeoutMs: Duration = 500.millis

val MaxRetryAttempts: Int = 2

val client = Http.client

.withLabel(modelPath)

.withMutualTls(serviceIdentifier)

.withRequestTimeout(MaxPredictionTimeoutMs)

.withTransport.connectTimeout(ConnectTimeoutMs)

.withSession.acquisitionTimeout(AcquisitionTimeoutMs)

.withHttpStats

val channel: ManagedChannel = FinagleChannelBuilder

.forTarget(modelPath)

.overrideAuthority("rustserving")

.maxRetryAttempts(MaxRetryAttempts)

.enableRetryForStatus(io.grpc.Status.RESOURCE\_EXHAUSTED)

.enableRetryForStatus(io.grpc.Status.UNKNOWN)

.enableUnsafeFullyBufferingMode()

.httpClient(client)

.build()

new PredictionGRPCService(channel)

}

}