package com.twitter.home\_mixer.module

import com.google.inject.Provides

import com.twitter.conversions.DurationOps.\_

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

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import com.twitter.inject.annotations.Flag

import com.twitter.timelinemixer.clients.feedback.FeedbackHistoryManhattanClient

import com.twitter.timelinemixer.clients.feedback.FeedbackHistoryManhattanClientConfig

import com.twitter.timelines.clients.manhattan.mhv3.ManhattanClientBuilder

import com.twitter.util.Duration

import javax.inject.Singleton

object FeedbackHistoryClientModule extends TwitterModule {

private val ProdDataset = "feedback\_history"

private val StagingDataset = "feedback\_history\_nonprod"

private final val Timeout = "mh\_feedback\_history.timeout"

flag[Duration](Timeout, 150.millis, "Timeout per request")

@Provides

@Singleton

def providesFeedbackHistoryClient(

@Flag(Timeout) timeout: Duration,

serviceId: ServiceIdentifier,

statsReceiver: StatsReceiver

) = {

val manhattanDataset = serviceId.environment.toLowerCase match {

case "prod" => ProdDataset

case \_ => StagingDataset

}

val config = new FeedbackHistoryManhattanClientConfig {

val dataset = manhattanDataset

val isReadOnly = true

val serviceIdentifier = serviceId

override val defaultMaxTimeout = timeout

}

new FeedbackHistoryManhattanClient(

ManhattanClientBuilder.buildManhattanEndpoint(config, statsReceiver),

manhattanDataset,

statsReceiver

)

}

}