package com.twitter.graph\_feature\_service.server

import com.google.inject.Module

import com.twitter.finatra.decider.modules.DeciderModule

import com.twitter.finatra.mtls.thriftmux.Mtls

import com.twitter.finatra.thrift.ThriftServer

import com.twitter.finatra.thrift.filters.{

AccessLoggingFilter,

LoggingMDCFilter,

StatsFilter,

ThriftMDCFilter,

TraceIdMDCFilter

}

import com.twitter.finatra.mtls.thriftmux.modules.MtlsThriftWebFormsModule

import com.twitter.finatra.thrift.routing.ThriftRouter

import com.twitter.graph\_feature\_service.server.controllers.ServerController

import com.twitter.graph\_feature\_service.server.handlers.ServerWarmupHandler

import com.twitter.graph\_feature\_service.server.modules.{

GetIntersectionStoreModule,

GraphFeatureServiceWorkerClientsModule,

ServerFlagsModule

}

import com.twitter.graph\_feature\_service.thriftscala

import com.twitter.inject.thrift.modules.ThriftClientIdModule

object Main extends ServerMain

class ServerMain extends ThriftServer with Mtls {

override val name = "graph\_feature\_service-server"

override val modules: Seq[Module] = {

Seq(

ServerFlagsModule,

DeciderModule,

ThriftClientIdModule,

GraphFeatureServiceWorkerClientsModule,

GetIntersectionStoreModule,

new MtlsThriftWebFormsModule[thriftscala.Server.MethodPerEndpoint](this)

)

}

override def configureThrift(router: ThriftRouter): Unit = {

router

.filter[LoggingMDCFilter]

.filter[TraceIdMDCFilter]

.filter[ThriftMDCFilter]

.filter[AccessLoggingFilter]

.filter[StatsFilter]

.add[ServerController]

}

override protected def warmup(): Unit = {

handle[ServerWarmupHandler]()

}

}