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

import com.google.inject.Module

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

import com.twitter.finatra.gizmoduck.modules.TimerModule

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

import com.twitter.finatra.thrift.ThriftServer

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

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.thriftscala

import com.twitter.graph\_feature\_service.worker.controllers.WorkerController

import com.twitter.graph\_feature\_service.worker.handlers.WorkerWarmupHandler

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

GraphContainerProviderModule,

WorkerFlagModule

}

import com.twitter.graph\_feature\_service.worker.util.GraphContainer

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

import com.twitter.util.Await

object Main extends WorkerMain

class WorkerMain extends ThriftServer with Mtls {

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

override val modules: Seq[Module] = {

Seq(

WorkerFlagModule,

DeciderModule,

TimerModule,

ThriftClientIdModule,

GraphContainerProviderModule,

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

)

}

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

router

.filter[LoggingMDCFilter]

.filter[TraceIdMDCFilter]

.filter[ThriftMDCFilter]

.filter[StatsFilter]

.add[WorkerController]

}

override protected def warmup(): Unit = {

val graphContainer = injector.instance[GraphContainer]

Await.result(graphContainer.warmup)

handle[WorkerWarmupHandler]()

}

}