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

import com.twitter.finatra.thrift.routing.ThriftWarmup

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

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

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

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

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

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

import com.twitter.inject.utils.Handler

import com.twitter.scrooge.Request

import com.twitter.util.logging.Logger

import javax.inject.Inject

import javax.inject.Singleton

import scala.util.Random

@Singleton

class ServerWarmupHandler @Inject() (warmup: ThriftWarmup) extends Handler {

val logger: Logger = Logger("WarmupHandler")

// TODO: Add the testing accounts to warm-up the service.

private val testingAccounts: Array[Long] = Seq.empty.toArray

private def getRandomRequest: GfsIntersectionRequest = {

GfsIntersectionRequest(

testingAccounts(Random.nextInt(testingAccounts.length)),

testingAccounts,

Seq(FeatureType(Following, FollowedBy), FeatureType(Following, FavoritedBy))

)

}

override def handle(): Unit = {

warmup.sendRequest(

GetIntersection,

Request(

GetIntersection.Args(

getRandomRequest

)),

10

)()

logger.info("Warmup Done!")

}

}