package com.twitter.cr\_mixer

import com.twitter.finagle.thrift.ClientId

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

import com.twitter.inject.Logging

import com.twitter.inject.utils.Handler

import com.twitter.product\_mixer.core.{thriftscala => pt}

import com.twitter.cr\_mixer.{thriftscala => st}

import com.twitter.scrooge.Request

import com.twitter.scrooge.Response

import com.twitter.util.Return

import com.twitter.util.Throw

import com.twitter.util.Try

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class CrMixerThriftServerWarmupHandler @Inject() (warmup: ThriftWarmup)

extends Handler

with Logging {

private val clientId = ClientId("thrift-warmup-client")

def handle(): Unit = {

val testIds = Seq(1, 2, 3)

try {

clientId.asCurrent {

testIds.foreach { id =>

val warmupReq = warmupQuery(id)

info(s"Sending warm-up request to service with query: $warmupReq")

warmup.sendRequest(

method = st.CrMixer.GetTweetRecommendations,

req = Request(st.CrMixer.GetTweetRecommendations.Args(warmupReq)))(assertWarmupResponse)

}

}

} catch {

case e: Throwable =>

// we don't want a warmup failure to prevent start-up

error(e.getMessage, e)

}

info("Warm-up done.")

}

private def warmupQuery(userId: Long): st.CrMixerTweetRequest = {

val clientContext = pt.ClientContext(

userId = Some(userId),

guestId = None,

appId = Some(258901L),

ipAddress = Some("0.0.0.0"),

userAgent = Some("FAKE\_USER\_AGENT\_FOR\_WARMUPS"),

countryCode = Some("US"),

languageCode = Some("en"),

isTwoffice = None,

userRoles = None,

deviceId = Some("FAKE\_DEVICE\_ID\_FOR\_WARMUPS")

)

st.CrMixerTweetRequest(

clientContext = clientContext,

product = st.Product.Home,

productContext = Some(st.ProductContext.HomeContext(st.HomeContext())),

)

}

private def assertWarmupResponse(

result: Try[Response[st.CrMixer.GetTweetRecommendations.SuccessType]]

): Unit = {

// we collect and log any exceptions from the result.

result match {

case Return(\_) => // ok

case Throw(exception) =>

warn("Error performing warm-up request.")

error(exception.getMessage, exception)

}

}

}