package com.twitter.tweetypie.util

import com.twitter.finagle.stats.Stat

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.servo

import com.twitter.util.Return

import com.twitter.util.Throw

import com.twitter.stitch.Stitch

object StitchUtils {

def trackLatency[T](latencyStat: Stat, s: => Stitch[T]): Stitch[T] = {

Stitch

.time(s)

.map {

case (res, duration) =>

latencyStat.add(duration.inMillis)

res

}

.lowerFromTry

}

def observe[T](statsReceiver: StatsReceiver, apiName: String): Stitch[T] => Stitch[T] = {

val stats = statsReceiver.scope(apiName)

val requests = stats.counter("requests")

val success = stats.counter("success")

val latencyStat = stats.stat("latency\_ms")

val exceptionCounter =

new servo.util.ExceptionCounter(stats, "failures")

stitch =>

trackLatency(latencyStat, stitch)

.respond {

case Return(\_) =>

requests.incr()

success.incr()

case Throw(e) =>

exceptionCounter(e)

requests.incr()

}

}

def translateExceptions[T](

stitch: Stitch[T],

translateException: PartialFunction[Throwable, Throwable]

): Stitch[T] =

stitch.rescue {

case t if translateException.isDefinedAt(t) =>

Stitch.exception(translateException(t))

case t => Stitch.exception(t)

}

}