package com.twitter.tweetypie.serverutil

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.servo

import com.twitter.servo.util.ExceptionCategorizer

object ExceptionCounter {

// These throwables are alertable because they indicate conditions we never expect in production.

def isAlertable(throwable: Throwable): Boolean =

throwable match {

case e: RuntimeException => true

case e: Error => true

case \_ => false

}

// count how many exceptions are alertable and how many are boring

val tweetypieCategorizers: ExceptionCategorizer =

ExceptionCategorizer.const("alertableException").onlyIf(isAlertable) ++

ExceptionCategorizer.const("boringException").onlyIf(BoringStackTrace.isBoring)

val defaultCategorizer: ExceptionCategorizer =

ExceptionCategorizer.default() ++ tweetypieCategorizers

def defaultCategorizer(name: String): ExceptionCategorizer =

ExceptionCategorizer.default(Seq(name)) ++ tweetypieCategorizers

def apply(statsReceiver: StatsReceiver): servo.util.ExceptionCounter =

new servo.util.ExceptionCounter(statsReceiver, defaultCategorizer)

def apply(statsReceiver: StatsReceiver, name: String): servo.util.ExceptionCounter =

new servo.util.ExceptionCounter(statsReceiver, defaultCategorizer(name))

def apply(

statsReceiver: StatsReceiver,

categorizer: ExceptionCategorizer

): servo.util.ExceptionCounter =

new servo.util.ExceptionCounter(statsReceiver, categorizer)

}