package com.github.lovasoa.superlogger;

public class App {

public static void main(String[] args) {

System.out.println("Agent started");

while (true) {

try {

System.out.println(new SystemInfo());

Thread.sleep(1000);

} catch (Throwable e) {

System.err.println(e);

}

}

}

}

package com.github.lovasoa.superlogger;

import com.sun.management.OperatingSystemMXBean;

import java.lang.management.ManagementFactory;

import java.net.InetAddress;

import java.net.UnknownHostException;

import javax.json.\*;

public class SystemInfo

{

OperatingSystemMXBean osBean;

public SystemInfo() {

osBean = ManagementFactory.getPlatformMXBean(OperatingSystemMXBean.class);

}

public double cpu() {

return osBean.getSystemCpuLoad();

}

public double memory() {

return (double) osBean.getFreePhysicalMemorySize() /

osBean.getTotalPhysicalMemorySize();

}

public String computerName() {

try {

return InetAddress.getLocalHost().getHostName();

} catch (UnknownHostException ex) {

return "unknown";

}

}

public String ip() {

try {

return InetAddress.getLocalHost().getHostAddress();

} catch (UnknownHostException ex) {

return "unknown";

}

}

public String toString() {

return Json.createObjectBuilder()

.add("cpu", cpu())

.add("memory", memory())

.add("computer", computerName())

.add("ip", ip())

.build()

.toString();

}

}

package com.github.lovasoa.superlogger;

import java.io.\*;

import java.net.\*;

class TCPServer

{

static int PORT = 6879;

public static void main(String argv[]) throws Exception

{

ServerSocket socket = new ServerSocket(PORT);

while (true) {

System.err.println("Waiting for connections on port " + PORT);

Socket conn = socket.accept();

System.err.println("New connection");

DataOutputStream out = new DataOutputStream(conn.getOutputStream());

boolean isOpen = true;

SystemInfo info = new SystemInfo();

String response;

while (isOpen) {

try {

out.writeBytes(info.toString() + "\n");

} catch(SocketException e) {

System.err.println(e);

isOpen = false;

}

Thread.sleep(1000);

}

System.err.println("Connection closed");

}

}

}