package com.helper;

import java.awt.AWTException;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.Socket;

import java.net.URL;

import java.net.URLConnection;

import java.net.UnknownHostException;

import java.util.Date;

import java.util.Enumeration;

import java.util.HashMap;

import java.io.InputStream;

import java.io.InputStreamReader;

/\*

\*/

public class StringHelper {

public static void main(String[] args) throws AWTException {

// Robot robot = new Robot();

// robot.mousePress(KeyEvent.BUTTON3\_MASK);

// robot.mouseRelease(KeyEvent.BUTTON3\_MASK);

// robot.mousePress(KeyEvent.BUTTON3\_MASK);

// robot.mouseRelease(KeyEvent.BUTTON3\_MASK);

try {

Socket s=new Socket("localhost",13);

// System.out.println(Inet4Address.getByName("www.googlse.com").toString());;

// Socket soc=new Socket("192.168.0.101", 9982);

} catch (UnknownHostException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static StringBuffer connect2Server(String url) {

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

URL u;

StringBuffer sb = new StringBuffer();

try {

u = new URL(url);

URLConnection uc = u.openConnection();

uc.setConnectTimeout(5000);

InputStream is = uc.getInputStream();

byte[] b = new byte[1024];

int i = 0;

while ((i = is.read(b)) != -1) {

String s = new String(b);

sb.append(s.trim());

}

u = null;

is.close();

} catch (Exception e1) {

e1.printStackTrace();

}

return sb;

}

public static String n2s(String d){

String dual="";

if(d==null){

dual = "";

}

else

dual=d.toString().trim();

return dual;

}

public static boolean n2b(Object d){

boolean dual=false;

if(d==null){

dual = false;

}

else

dual=new Boolean(d.toString()).booleanValue();

return dual;

}

public static String n2s(Object d){

String dual="";

if(d==null){

dual = "";

}

else

dual=d.toString().trim();

return dual;

}

public static float nullObjectToFloatEmpty(Object d){

float i=0;

if(d!=null){

String dual=d.toString().trim();

try{

i=new Float(dual).floatValue();

}catch (Exception e) {

System.out.println("Unable to find integer value");

}

}

return i;

}

public static double n2d(Object d){

double i=0;

if(d!=null){

String dual=d.toString().trim();

try{

i=new Double(dual).doubleValue();

}catch (Exception e) {

System.out.println("Unable to find integer value");

}

}

return i;

}

public static float n2f(Object d){

float i=0;

if(d!=null){

String dual=d.toString().trim();

try{

i=new Float(dual).floatValue();

}catch (Exception e) {

System.out.println("Unable to find integer value");

}

}

return i;

}

public static int n2i(Object d){

int i=0;

if(d!=null){

String dual=d.toString().trim();

try{

i=new Integer(dual).intValue();

}catch (Exception e) {

System.out.println("Unable to find integer value");

}

}

return i;

}

// public static HashMap displayRequest(ServletRequest request){

//

// Enumeration paraNames = request.getParameterNames();

//

// System. out.println(" ------------------------------");

//

// System. out.println("parameters:");

//

// HashMap parameters=new HashMap();

//

//

// String pname;

//

// String pvalue;

//

// while (paraNames.hasMoreElements()) {

//

// pname = (String)paraNames.nextElement();

//

// pvalue = request.getParameter(pname);

//

// System.out.println(pname + " = " + pvalue + "");

//

// parameters.put(pname, pvalue);

// }

//

// Enumeration<String> requestAttributes = request.getAttributeNames();

// while (requestAttributes.hasMoreElements()) {

// try{

// String attributeName = n2s(requestAttributes.nextElement());

// String attributeValue = n2s(request.getAttribute(attributeName));

//

// parameters.put(attributeName, attributeValue);

// }catch (Exception e) {

// // TODO: handle exception

// }

//

// }

//

// System.out.println(" ------------------------------");

// return parameters;

// }

public static StringBuffer readURLContent(String url) {

System.out.println("URL " + url);

StringBuffer json = new StringBuffer();

try {

BufferedReader in = new BufferedReader(new InputStreamReader(

new URL(url).openStream()));

String line;

while ((line = in.readLine()) != null) {

json.append(line);

}

} catch (Exception e) {

e.printStackTrace();

}

return json;

}

}