NEW WEB SERVICE

package blind;

import com.commondb.Common\_DB;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

/\*\*

\*

\* @author Android

\*/

@WebService(serviceName = "NewWebService")

public class NewWebService {

/\*\*

\* This is a sample web service operation

\*/

@WebMethod(operationName = "Register")

public String Register(@WebParam(name = "username") String username,@WebParam(name = "type") String type) {

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/blindapp", "root","password");

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("select \* from register where username='"+username+"'");

if(rs.next())

{

String status=rs.getString("status");

int count=Integer.parseInt(status)+1;

String result=String.valueOf(count);

PreparedStatement ps=con.prepareStatement("update register set status='"+result+"' where username='"+username+"'");

ps.executeUpdate();

}

else

{

PreparedStatement ps=con.prepareStatement("insert into register(username,type) values ('"+username+"','"+type+"')");

ps.executeUpdate();

}

}

catch(ClassNotFoundException | SQLException e)

{

System.out.println(e);

}

return type;

}

@WebMethod(operationName = "LanguageLearn")

public String LanguageLearn(@WebParam(name = "username") String username,@WebParam(name = "language") String language) {

System.out.println("Username: "+username);

System.out.println("Language: "+language);

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/blindapp", "root","password");

PreparedStatement ps=con.prepareStatement("update register set language1='"+language+"' where username='"+username+"'");

ps.executeUpdate();

}

catch(ClassNotFoundException | SQLException e)

{

System.out.println(e);

}

return language;

}

@WebMethod(operationName = "getDeviceList1")

public String getDeviceList1(@WebParam(name = "report") String report) {

String braille;

StringBuilder sb = new StringBuilder();

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/blindapp","root","password");

Statement st=con.createStatement();

ResultSet rs1=st.executeQuery("select \* from braille");

if(rs1.next()){

braille=rs1.getString("letters");

System.out.println("Braille Value: "+braille);

return braille;

//

//String sqlquery="DROP TABLE braille";

}

else

{

return "Sorry Wrong Braille";

}

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(NewWebService.class.getName()).log(Level.SEVERE, null, ex);

}

try {

} catch (Exception ex) {

Logger.getLogger(NewWebService.class.getName()).log(Level.SEVERE, null, ex);

}

//return sb.toString();

// try

// {

// Class.forName("com.mysql.jdbc.Driver");

// Connection con1=DriverManager.getConnection("jdbc:mysql://localhost:3306/blindapp","root","password");

// PreparedStatement ps=con1.prepareStatement("TRUNCATE TABLE braille");

// ps.executeUpdate();

// }

// catch(Exception e)

// {

// System.out.println(e);

// }

return null;

}

}

SERVLET PROGRAM

package blind;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\*

\* @author Admin

\*/

@WebServlet(name = "NewServlet", urlPatterns = {"/NewServlet"})

public class NewServlet extends HttpServlet {

/\*\*

\* Processes requests for both HTTP <code>GET</code> and <code>POST</code>

\* methods.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

/\* TODO output your page here. You may use following sample code. \*/

String a=request.getParameter("a");

System.out.println("Letter is: "+a);

out.println("Success");

}

}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

/\*\*

\* Handles the HTTP <code>GET</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Handles the HTTP <code>POST</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Returns a short description of the servlet.

\*

\* @return a String containing servlet description

\*/

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}

import com.commondb.Common\_DB;

import java.io.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.\*; //import gnu.io.\*;

import javax.comm.\*;

public class SimpleRead implements Runnable, SerialPortEventListener {

static CommPortIdentifier portId;

static Enumeration portList;

InputStream inputStream;

SerialPort serialPort;

Thread readThread;

byte[] readBuffer;

int z=0;

String data=null;

public String recvdData="aa";

public static void main(String[] args) {

portList = CommPortIdentifier.getPortIdentifiers();

System.out.println("portList... " + portList);

while (portList.hasMoreElements()) {

portId = (CommPortIdentifier) portList.nextElement();

if (portId.getPortType() == CommPortIdentifier.PORT\_SERIAL) {

System.out.println("port identified is Serial.. "

+ portId.getPortType());

if (portId.getName().equals("COM5")) {

System.out.println("port identified is COM5.. "

+ portId.getName());

// if (portId.getName().equals("/dev/term/a")) {

SimpleRead reader = new SimpleRead();

} else {

System.out.println("unable to open port");

}

}

}

}

public SimpleRead() {

try {

System.out.println("In SimpleRead() contructor");

serialPort = (SerialPort) portId.open("SimpleReadApp1111",2000);

System.out.println(" Serial Port.. " + serialPort);

} catch (PortInUseException e) {

System.out.println("Port in use Exception");

}

try {

inputStream = serialPort.getInputStream();

System.out.println(" Input Stream... " + inputStream);

} catch (IOException e) {

System.out.println("IO Exception");

}

try {

serialPort.addEventListener(this);

} catch (TooManyListenersException e) {

System.out.println("Tooo many Listener exception");

}

serialPort.notifyOnDataAvailable(true);

try {

serialPort.setSerialPortParams(9600, SerialPort.DATABITS\_8,

SerialPort.STOPBITS\_1, SerialPort.PARITY\_NONE);

// no handshaking or other flow control

serialPort.setFlowControlMode(SerialPort.FLOWCONTROL\_NONE);

// timer on any read of the serial port

serialPort.enableReceiveTimeout(500);

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

} catch (UnsupportedCommOperationException e) {

System.out.println("UnSupported comm operation");

}

readThread = new Thread(this);

readThread.start();

}

public void run() {

try {

System.out.println("In run() function ");

Thread.sleep(100);

// System.out.println();

} catch (InterruptedException e) {

System.out.println("Interrupted Exception in run() method");

}

}

public void serialEvent(SerialPortEvent event) {

switch (event.getEventType())

{

case SerialPortEvent.DATA\_AVAILABLE:

// System.out.println("DATA\_AVAILABLE");

byte[] readBuffer = new byte[1024];

int numBytes=1024;

data="";

String x="";

// Connection con=null;

// Statement st=null;

// ResultSet rs=null;

try

{

Thread.sleep(100);

while (inputStream.available() > 0)

{

numBytes = inputStream.read(readBuffer);//count of reading data

data=data+new String(readBuffer,0,numBytes);

data=data.trim();

this.recvdData+=data;

}

System.out.println("Your data is: "+data);

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/blindapp","root","password");

PreparedStatement ps=con.prepareStatement("insert into braille values(?)");

ps.setString(1, data);

ps.executeUpdate();

System.out.println("Successfully Stored");

}

catch(ClassNotFoundException | SQLException e)

{

System.out.println(e);

}

}

catch (InterruptedException | IOException e)

{

System.out.println("Exception in serial event-->"+e);

}

break;//break from switch case 1:

}//end of switch

}

}

SIMPLE READ PROGRAM

import com.commondb.Common\_DB;

import java.io.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.\*; //import gnu.io.\*;

import javax.comm.\*;

public class SimpleRead implements Runnable, SerialPortEventListener {

static CommPortIdentifier portId;

static Enumeration portList;

InputStream inputStream;

SerialPort serialPort;

Thread readThread;

byte[] readBuffer;

int z=0;

String data=null;

public String recvdData="aa";

public static void main(String[] args) {

portList = CommPortIdentifier.getPortIdentifiers();

System.out.println("portList... " + portList);

while (portList.hasMoreElements()) {

portId = (CommPortIdentifier) portList.nextElement();

if (portId.getPortType() == CommPortIdentifier.PORT\_SERIAL) {

System.out.println("port identified is Serial.. "

+ portId.getPortType());

if (portId.getName().equals("COM5")) {

System.out.println("port identified is COM5.. "

+ portId.getName());

// if (portId.getName().equals("/dev/term/a")) {

SimpleRead reader = new SimpleRead();

} else {

System.out.println("unable to open port");

}

}

}

}

public SimpleRead() {

try {

System.out.println("In SimpleRead() contructor");

serialPort = (SerialPort) portId.open("SimpleReadApp1111",2000);

System.out.println(" Serial Port.. " + serialPort);

} catch (PortInUseException e) {

System.out.println("Port in use Exception");

}

try {

inputStream = serialPort.getInputStream();

System.out.println(" Input Stream... " + inputStream);

} catch (IOException e) {

System.out.println("IO Exception");

}

try {

serialPort.addEventListener(this);

} catch (TooManyListenersException e) {

System.out.println("Tooo many Listener exception");

}

serialPort.notifyOnDataAvailable(true);

try {

serialPort.setSerialPortParams(9600, SerialPort.DATABITS\_8,

SerialPort.STOPBITS\_1, SerialPort.PARITY\_NONE);

// no handshaking or other flow control

serialPort.setFlowControlMode(SerialPort.FLOWCONTROL\_NONE);

// timer on any read of the serial port

serialPort.enableReceiveTimeout(500);

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

} catch (UnsupportedCommOperationException e) {

System.out.println("UnSupported comm operation");

}

readThread = new Thread(this);

readThread.start();

}

public void run() {

try {

System.out.println("In run() function ");

Thread.sleep(100);

// System.out.println();

} catch (InterruptedException e) {

System.out.println("Interrupted Exception in run() method");

}

}

public void serialEvent(SerialPortEvent event) {

switch (event.getEventType())

{

case SerialPortEvent.DATA\_AVAILABLE:

// System.out.println("DATA\_AVAILABLE");

byte[] readBuffer = new byte[1024];

int numBytes=1024;

data="";

String x="";

// Connection con=null;

// Statement st=null;

// ResultSet rs=null;

try

{

Thread.sleep(100);

while (inputStream.available() > 0)

{

numBytes = inputStream.read(readBuffer);//count of reading data

data=data+new String(readBuffer,0,numBytes);

data=data.trim();

this.recvdData+=data;

}

System.out.println("Your data is: "+data);

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/blindapp","root","password");

PreparedStatement ps=con.prepareStatement("insert into braille values(?)");

ps.setString(1, data);

ps.executeUpdate();

System.out.println("Successfully Stored");

}

catch(ClassNotFoundException | SQLException e)

{

System.out.println(e);

}

}

catch (InterruptedException | IOException e)

{

System.out.println("Exception in serial event-->"+e);

}

break;//break from switch case 1:

}//end of switch

}

}