package in.cdac.hash;

import java.io.IOException;

import java.io.PrintWriter;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class PullSecureServlet

\*/

public class PullSecureServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public PullSecureServlet() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse

\* response)

\*/

protected void doGet(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse

\* response)

\*/

protected void doPost(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

String finalresponse = "not Authenticated";

PrintWriter out = response.getWriter();

/\*

\* get the random number provided to department at the time of

\* registration.

\*/

String RandomNumber = "here get the random number provided to department at the time of registration.";

String MobileNo = ESAPI.encoder().encodeForHTML(request.getParameter("mobileNumber"));

String TimeStamp = ESAPI.encoder().encodeForHTML(request.getParameter("timeStamp"));

String OperatorName = ESAPI.encoder().encodeForHTML(request.getParameter("operatorName"));

String Areacode = ESAPI.encoder().encodeForHTML(request.getParameter("areaCode"));

String Message = ESAPI.encoder().encodeForHTML(request.getParameter("message"));

String recievedHash = ESAPI.encoder().encodeForHTML(request.getParameter("hash"));

System.out.println("MobileNo==" + MobileNo + "\n" + "TimeStamp=="

+ TimeStamp + "\n" + "OperatorName==" + OperatorName + "\n"

+ "AreaCode==" + Areacode + "\n" + "Message==" + Message + "\n"

+ "hash==" + recievedHash);

String hashValue = hashGenerate(TimeStamp, Message, MobileNo,

RandomNumber);

System.out.println("final generated hash is ::;" + hashValue);

if (hashValue.equals(recievedHash)) {

System.out

.println("both hash are equal and request is from authenticated source.");

finalresponse="if Authenticated provide your response";

} else {

System.out.println("both hash are different");

finalresponse="if not Authenticated provide your response";

}

out.println(finalresponse);

}

protected String hashGenerate(String timestamp, String message,

String mobileNo, String randomNumber) {

StringBuffer hexString = null;

StringBuffer finalString = new StringBuffer();

finalString.append(mobileNo.trim()).append(timestamp.trim())

.append(message.trim()).append(randomNumber.trim());

System.out.println("final string is ::" + finalString.toString());

String hashGen = finalString.toString();

MessageDigest md;

try {

md = MessageDigest.getInstance("SHA-512");

md.update(hashGen.getBytes());

byte byteData[] = md.digest();

// convert the byte to hex format method 1

StringBuffer sb = new StringBuffer();

for (int i = 0; i < byteData.length; i++) {

sb.append(Integer.toString((byteData[i] & 0xff) + 0x100, 16)

.substring(1));

}

//System.out.println("Hex format : " + sb.toString());

// convert the byte to hex format method 2

hexString = new StringBuffer();

for (int i = 0; i < byteData.length; i++) {

String hex = Integer.toHexString(0xff & byteData[i]);

if (hex.length() == 1)

hexString.append('0');

hexString.append(hex);

}

//System.out.println("Hex format : " + hexString.toString());

} catch (NoSuchAlgorithmException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return hexString.toString();

}

}