**CODE:**

**DBConnection.java**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package network;

/\*\*

\*

\* @author java3

\*/

import java.sql.Connection;

import java.sql.DriverManager;

/\*\*

\*

\* @suresh java3

\*/

public class DbConnection {

public static Connection getConnection()

{

Connection con = null;

try{

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

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

}

catch(Exception e)

{

e.printStackTrace();

}

return con;

}

}

**fileUpload:**

package network;

import com.oreilly.servlet.MultipartRequest;

import com.sun.org.apache.xerces.internal.impl.dv.util.Base64;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

/\*\*

\*

\* @author IBN5

\*/

public class UploadFiles extends HttpServlet {

File file;

final String filepath = "D:/";

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

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

PrintWriter out = response.getWriter();

try {

MultipartRequest m = new MultipartRequest(request, filepath);

String pkey = m.getParameter("pkey");

String fname = m.getParameter("fname");

String des = m.getParameter("des");

String accesspolicy = m.getParameter("ap");

File file = m.getFile("file");

String filename = file.getName().toLowerCase();

Connection con = DbConnection.getConnection();

BufferedReader br = new BufferedReader(new FileReader(filepath + filename));

StringBuffer sb = new StringBuffer();

String temp = null;

while ((temp = br.readLine()) != null) {

sb.append(temp);

}

KeyGenerator keyGen = KeyGenerator.getInstance("AES");

keyGen.init(128);

SecretKey secretKey = keyGen.generateKey();

System.out.println("secret key:" + secretKey);

Encryption e = new Encryption();

String CipherText = e.encrypt(sb.toString(), secretKey);

FileWriter fw = new FileWriter(file);

fw.write(CipherText);

fw.close();

byte[] b = secretKey.getEncoded();

String skey = Base64.encode(b);

System.out.println("converted secretkey to string:" + skey);

HttpSession user = request.getSession(true);

String oid = (String) user.getAttribute("oid");

String owner = user.getAttribute("oname").toString();

DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

Date date = new Date();

String time = dateFormat.format(date);

// boolean status = new Ftpcon().upload(file);

//if (status) {

Statement st = con.createStatement();

int i = st.executeUpdate("insert into fileupload(filename,content,owner,time,secret\_key,public\_key, oid, des, accesspolicy)values('" + file.getName() + "','" + CipherText + "','" + owner + "','" + time + "','" + skey + "','" + pkey + "', '"+oid+"','"+des+"','"+accesspolicy+"')");

System.out.println(i);

if (i != 0) {

response.sendRedirect("uploadfile.jsp?msg=File\_upload successfully");

} else {

out.println("error while uploading additional informations");

}

//} else {

// out.println("error");

// }

} catch (Exception e) {

out.println(e);

} finally {

out.close();

}

}

// <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>

}

**Encryption Algorithm**:

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.app.utility;

import com.app.action.\*;

import java.security.Key;

import javax.crypto.Cipher;

import javax.crypto.spec.SecretKeySpec;

import sun.misc.BASE64Decoder;

import sun.misc.BASE64Encoder;

public class EncryptionAlgoritham {

private static final String ALGO = "AES";

private static final byte[] keyValue =

new byte[] { 'T', 'h', 'e', 'B', 'e', 's', 't','S', 'e', 'c', 'r','e', 't', 'K', 'e', 'y' };

public static String encrypt(String Data) throws Exception {

Key key = generateKey();

Cipher c = Cipher.getInstance(ALGO);

c.init(Cipher.ENCRYPT\_MODE, key);

byte[] encVal = c.doFinal(Data.getBytes());

String encryptedValue = new BASE64Encoder().encode(encVal);

return encryptedValue;

}

public static String decrypt(String encryptedData) throws Exception {

Key key = generateKey();

Cipher c = Cipher.getInstance(ALGO);

c.init(Cipher.DECRYPT\_MODE, key);

byte[] decordedValue = new BASE64Decoder().decodeBuffer(encryptedData);

byte[] decValue = c.doFinal(decordedValue);

String decryptedValue = new String(decValue);

return decryptedValue;

}

private static Key generateKey() throws Exception {

Key key = new SecretKeySpec(keyValue, ALGO);

return key;

}

}

**Decryption.java**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package network;

/\*\*

\*

\* @author java3

\*/

import com.sun.org.apache.xerces.internal.impl.dv.util.Base64;

import java.io.ByteArrayOutputStream;

import java.io.FileInputStream;

import java.io.FileWriter;

import java.util.Scanner;

import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import javax.crypto.spec.SecretKeySpec;

import javax.swing.JOptionPane;

import sun.misc.BASE64Decoder;

import sun.misc.BASE64Encoder;

public class Decryption

{

public String decrypt(String txt,String skey)

{

String decryptedtext = null;

try

{

//converting string to secretkey

byte[] bs=Base64.decode(skey);

SecretKey sec=new SecretKeySpec(bs, "AES");

System.out.println("converted string to seretkey:"+sec);

System.out.println("secret key:"+sec);

Cipher aesCipher = Cipher.getInstance("AES");//getting AES instance

aesCipher.init(Cipher.ENCRYPT\_MODE,sec);//initiating ciper encryption using secretkey

byte[] byteCipherText =new BASE64Decoder().decodeBuffer(txt); //encrypting data

// System.out.println("ciper text:"+byteCipherText);

aesCipher.init(Cipher.DECRYPT\_MODE,sec,aesCipher.getParameters());//initiating ciper decryption

byte[] byteDecryptedText = aesCipher.doFinal(byteCipherText);

decryptedtext = new String(byteDecryptedText);

System.out.println("Decrypted Text:"+decryptedtext);

}

catch(Exception e)

{

System.out.println(e);

}

return decryptedtext;

}

}

**CiperText.java:**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package network;

/\*\*

\*

\* @author java4

\*/

import java.io.InputStream;

import java.security.spec.KeySpec;

import javax.crypto.Cipher;

import javax.crypto.SecretKey;

import javax.crypto.SecretKeyFactory;

import javax.crypto.spec.DESedeKeySpec;

import org.apache.tomcat.util.codec.binary.Base64;

public class CiperText {

private static final String UNICODE\_FORMAT = "UTF8";

public static final String DESEDE\_ENCRYPTION\_SCHEME = "DESede";

private KeySpec ks;

private SecretKeyFactory skf;

private Cipher cipher;

byte[] arrayBytes;

private String myEncryptionKey;

private String myEncryptionScheme;

SecretKey key;

public CiperText() throws Exception {

myEncryptionKey = "ThisIsSpartaThisIsSparta";

myEncryptionScheme = DESEDE\_ENCRYPTION\_SCHEME;

arrayBytes = myEncryptionKey.getBytes(UNICODE\_FORMAT);

ks = new DESedeKeySpec(arrayBytes);

skf = SecretKeyFactory.getInstance(myEncryptionScheme);

cipher = Cipher.getInstance(myEncryptionScheme);

key = skf.generateSecret(ks);

}

public String encrypt(String unencryptedString) {

String encryptedString = null;

try {

cipher.init(Cipher.ENCRYPT\_MODE, key);

byte[] plainText = unencryptedString.getBytes(UNICODE\_FORMAT);

byte[] encryptedText = cipher.doFinal(plainText);

encryptedString = new String(Base64.encodeBase64(encryptedText));

} catch (Exception e) {

e.printStackTrace();

}

return encryptedString;

}

public String decrypt(String encryptedString) {

String decryptedText=null;

try {

cipher.init(Cipher.DECRYPT\_MODE, key);

byte[] encryptedText1 = encryptedString.getBytes(UNICODE\_FORMAT);

byte[] encryptedText = Base64.decodeBase64(encryptedText1);

byte[] plainText = cipher.doFinal(encryptedText);

decryptedText= new String(plainText);

} catch (Exception e) {

e.printStackTrace();

}

return decryptedText;

}

String encrypt(SecretKey secretKey) {

throw new UnsupportedOperationException("Not yet implemented");

}

}

**Publickeyrequest.jsp:**

<%--

Document : index

Created on : 19 Dec, 2020, 11:17:23 AM

Author : aruma

--%>

<%@page import="network.DbConnection"%>

<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.Statement"%>

<%@page import="java.sql.Connection"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- The above 3 meta tags \*must\* come first in the head; any other head content must come \*after\* these tags -->

<title>Fog-Computing</title>

<link rel="icon" href="img/fav.png" type="image/x-icon">

<!-- Bootstrap -->

<link href="css/bootstrap.min.css" rel="stylesheet">

<link href="ionicons/css/ionicons.min.css" rel="stylesheet">

<!-- main css -->

<link href="css/style.css" rel="stylesheet">

<!-- modernizr -->

<script src="js/modernizr.js"></script>

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->

<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body>

<!-- Preloader -->

<div id="preloader">

<div class="pre-container">

<div class="spinner">

<div class="double-bounce1"></div>

<div class="double-bounce2"></div>

</div>

</div>

</div>

<!-- end Preloader -->

<div class="container-fluid">

<!-- box header -->

<header class="box-header">

<div class="box-logo">

<!--<a href="index.html"><img src="img/logo.png" width="80" alt="Logo"></a>-->

</div>

<!-- box-nav -->

<a class="box-primary-nav-trigger" href="#0">

<span class="box-menu-text">Menu</span><span class="box-menu-icon"></span>

</a>

<!-- box-primary-nav-trigger -->

</header>

<!-- end box header -->

<!-- nav -->

<nav>

<ul class="box-primary-nav">

<!--<li class="box-label">About me</li>-->

<li><a href="ohome.jsp">Home</a></li>

<li><a href="pkreq.jsp">Public Key Request</a></li>

<li><a href="uploadfile.jsp">Upload File</a></li>

<li><a href="ovfilereq.jsp">View Data User File Access Request</a></li>

<li><a href="ouploadfiles.jsp">View Uploaded Files</a></li>

<li><a href="olog.jsp">Logout</a></li>

</ul>

</nav>

<!-- end nav -->

<!-- box-intro -->

<section class="box-intro">

<div class="table-cell">

<!--<h1 class="box-headline letters rotate-2">

<span class="box-words-wrapper">

<b class="is-visible">design.</b>

<b>&nbsp;coding.</b>

<b>graphic.</b>

</span>

</h1>--><h3 style="color: blue">Send Public key request</h3>

<h3 style="color: red">

<center> <table summary="Summary Here" cellpadding="0" cellspacing="0">

<thead>

<tr>

<th>Id</th>

<th>Name</th>

<th>Mail</th>

<th>Status</th>

<th>Give Request</th>

</tr>

</thead>

<%

String id = (String) session.getAttribute("oid");

System.out.println("uiddd for key====" + id);

Connection con = null;

Statement st = null;

ResultSet rs = null;

try {

con = DbConnection.getConnection();

st = con.createStatement();

rs = st.executeQuery("select \* from oreg where id ='" + id + "'");

while (rs.next()) {%>

<tr class="light" style="color:black">

<td><%=rs.getString("id")%>&nbsp;&nbsp;&nbsp;</td>

<td><%=rs.getString("name")%>&nbsp;&nbsp;&nbsp;</td>

<td><%=rs.getString("mail")%>&nbsp;&nbsp;&nbsp;</td>

<td><%=rs.getString("status")%>&nbsp;&nbsp;&nbsp;</td>

<td><a href="pkreq1.jsp?id=<%=rs.getString("id")%>"><font style="color: #e60498">Request</font></a></td>

</tr>

<%}

} catch (Exception ex) {

ex.printStackTrace();

}

%>

</table></center>

</div></h3>

</div>

<!--<div class="mouse">

<div class="scroll"></div>

</div>-->

</section>

<!-- end box-intro -->

</div>

<!-- portfolio div -->

<!-- footer -->

<footer>

<div class="container-fluid">

<p class="copyright">© Box Portfolio 2016</p>

</div>

</footer>

<!-- end footer -->

<!-- back to top -->

<a href="#0" class="cd-top"><i class="ion-android-arrow-up"></i></a>

<!-- end back to top -->

<!-- jQuery -->

<script src="js/jquery-2.1.1.js"></script>

<!-- plugins -->

<script src="js/bootstrap.min.js"></script>

<script src="js/menu.js"></script>

<script src="js/animated-headline.js"></script>

<script src="js/isotope.pkgd.min.js"></script>

<!-- custom script -->

<script src="js/custom.js"></script>

</body>

</html>