**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Jboss Basic Authentication\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

open D:\Setups\JBOSS 4.2 GA\server\default\conf

Edit users.PROPERTIES and roles.PROPERTIES

in users.PROPERTIES, write:

# Sample ResourceBundle properties file

# myuserid is a user id

# mypassword is a password assigned to myuserid

pravin=surve

**\*\*\*\*\*\*\*\*\*\*\*\*in roles.PROPERTIES, write:\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

# Sample ResourceBundle properties file

# myuserid is a user who?s assigned a role of admin

pravin=admin

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Open login-config.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

make following entry:

<application-policy name = "**my-web**">

<authentication>

<login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule"

flag = "required">

<module-option name="usersProperties">users.properties</module-option>

<module-option name="rolesProperties">roles.properties</module-option>

</login-module>

</authentication>

</application-policy>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*In Jboss-web.xml make following entry:\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE jboss-web

PUBLIC "-//JBoss//DTD Web Application 2.3V2//EN"

"http://www.jboss.org/j2ee/dtd/jboss-web\_3\_2.dtd">

<jboss-web>

<security-domain>java:/jaas/**my-web**</security-domain>

</jboss-web>

**\*\*\*\*\*\*In web.xml make following entry:\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<security-constraint>

<web-resource-collection>

<web-resource-name>All resources</web-resource-name>

<description>Protects all resources</description>

<url-pattern>/\*</url-pattern>

<http-method>GET</http-method>

<http-method>POST</http-method>

</web-resource-collection>

<auth-constraint>

<role-name>admin</role-name>

</auth-constraint>

</security-constraint>

<security-role>

<role-name>admin</role-name>

</security-role>

<login-config>

<auth-method>BASIC</auth-method>

<realm-name>Test Realm</realm-name>

</login-config>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*J2EE MPT Assignment\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*Controller \*\*\*\*\*\*\*\*\*\*\*\*ProcessUser.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**package** com.igate.controller;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.RequestDispatcher;

**import** javax.servlet.ServletConfig;

**import** javax.servlet.ServletException;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

**import** com.igate.dto.UserDto;

**import** com.igate.exception.UsersException;

**import** com.igate.service.UserService;

**public** **class** ProcessUser **extends** HttpServlet

{

**private** **static** **final** **long** *serialVersionUID* = 1L;

**public** ProcessUser() { **super**(); }

**public** **void** init(ServletConfig config) **throws** ServletException

{**super**.init(config);}

**public** **void** destroy(){ }

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException

{

System.*out*.println("In doget..................");

doPost(request,response);

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException

{

System.*out*.println("In doPost..................");

PrintWriter pw=response.getWriter();

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

System.*out*.println("In doPost.................."+action);

UserService userService = **new** UserService();

**if**(action!=**null**)

{

**if**(action.equals("Register"))

{

//Create a new HttpSession

HttpSession userSession=request.getSession(**true**);

UserDto userDTO = **new** UserDto();

userDTO.setName(request.getParameter("custName"));

userDTO.setMobilenumber(request.getParameter("mobileNo"));

userDTO.setUsername(request.getParameter("uName"));

userDTO.setPassword(request.getParameter("pass"));

//double bill = (double) Math.random();

**double** bill=(**double**)Math.*round*(Math.*random*()\*100);

userDTO.setBillamount(bill);

System.*out*.println(userDTO);

**try**

{

userService.insertUsers(userDTO);

}

**catch** (UsersException e)

{

System.*out*.println("User Exception in Controller Servlet....");

e.printStackTrace();

}

userSession.setAttribute("UserDetails",userDTO);

RequestDispatcher rd=request.getRequestDispatcher("customerHome.jsp");

rd.forward(request,response);

}

**if**(action.equals("PayBill"))

{

HttpSession useSession = request.getSession(**true**);

UserDto userDTO =(UserDto) useSession.getAttribute("UserDetails");

System.*out*.println(" Paybill action...."+userDTO);

RequestDispatcher rd=request.getRequestDispatcher("/jsps/payBill.jsp");

rd.forward(request,response);

}

**if**(action.equals("Pay"))

{

**int** amt=Integer.*parseInt*(request.getParameter("amount"));

String cno=request.getParameter("cardnumber");

System.*out*.println(" Pay action...."+amt);

request.setAttribute("AMOUNT",amt);

request.setAttribute("CNO", cno);

RequestDispatcher rd=request.getRequestDispatcher("/jsps/payBill.jsp");

rd.forward(request,response);

}

**if**(action.equals("Success"))

{

System.*out*.println(" Success action....");

Integer amt=(Integer)request.getAttribute("AMOUNT");

System.*out*.println(" Success action...."+amt.intValue());

HttpSession useSession = request.getSession(**true**);

UserDto userDTO =(UserDto) useSession.getAttribute("UserDetails");

**try**

{

**int** penAmt=userService.updateAmt(amt,userDTO);

request.setAttribute("PendingAmt", penAmt);

RequestDispatcher rd=request.getRequestDispatcher("jsps/success.jsp");

rd.forward(request,response);

}

**catch** (UsersException e)

{

System.*out*.println("User Exception in Controller Servlet....");

e.printStackTrace();

}

}

//Login.jsp

**if**(action.equals("Login"))

{

System.*out*.println(" Login action....");

**try**

{

request.setAttribute("UserNames", userService.getUsers());

RequestDispatcher rd=request.getRequestDispatcher("jsps/login.jsp");

rd.forward(request,response);

}

**catch** (Exception e)

{

System.*out*.println("User Exception in Controller Servlet....");

e.printStackTrace();

}

}

//ShowBill.jsp

**if**(action.equals("ShowBillAmt"))

{

String username=request.getParameter("uName");

String pwd=request.getParameter("pass");

System.*out*.println(" ShowBillAmt action...."+username);

**try**

{

UserDto userDto=userService.getBillAmt(username);

HttpSession useSession = request.getSession(**true**);

useSession.setAttribute("UserDetails",userDto);

System.*out*.println(" ShowBillAmt action...."+userDto);

RequestDispatcher rd=request.getRequestDispatcher("/jsps/payBill.jsp"); // 1

rd.forward(request,response);

}

**catch** (Exception e)

{

System.*out*.println("User Exception in Controller Servlet....");

e.printStackTrace();

}

}

}

**else**

{

System.*out*.println(" action is null ");

} }

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DAO UserDao.java \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.dao;

import java.util.ArrayList;

import java.util.HashMap;

import com.igate.dto.UserDto;

import com.igate.exception.UsersException;

public interface UserDao

{

public void insertUsers(UserDto userDTO) throws UsersException;

public int updateAmt(Integer amt,UserDto userDTO)throws UsersException;

public ArrayList<String> getUsers()throws UsersException;

public UserDto getBillAmt(String username)throws UsersException;

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* UserDaoImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.HashMap;

import com.igate.dto.UserDto;

import com.igate.exception.UsersException;

import com.igate.util.DbUtil;

public class UserDaoImpl implements UserDao

{

Connection con=null;

PreparedStatement pst1=null;

PreparedStatement pst2=null;

PreparedStatement pst3=null;

ResultSet rs=null;

ResultSet rs2=null;

@Override

public void insertUsers(UserDto userDTO) throws UsersException

{

con=DbUtil.getConnection();

System.out.println(" Conneted in insertUsers.............");

try

{

con.setAutoCommit(false);

pst1=con.prepareStatement("insert into users values(?,?,?,?)");

pst1.setString(1, userDTO.getName());

pst1.setString(2,userDTO.getUsername());

pst1.setString(3, userDTO.getPassword());

pst1.setString(4, userDTO.getMobilenumber());

int noOfRec1=pst1.executeUpdate();

System.out.println(" Data is inserted successfully in users table");

pst2=con.prepareStatement("insert into bills values(?,?)");

pst2.setString(1, userDTO.getMobilenumber());

pst2.setDouble(2,userDTO.getBillamount());

int noOfRec2=pst2.executeUpdate();

System.out.println(" Data is inserted successfully in bilss table");

con.commit();

}

catch (SQLException e)

{

try

{

con.rollback();

}

catch (SQLException e1)

{

throw new UsersException("some exception in insert User method"+e1.getMessage() );

}

throw new UsersException("some exception in insert User method"+e.getMessage() );

}

finally

{

try

{

pst1.close();

pst2.close();

con.close();

}

catch (SQLException e)

{

throw new UsersException("some exception while closing objects "+e.getMessage() );

}

}

}

@Override

public int updateAmt(Integer amt,UserDto userDTOt) throws UsersException

{

con=DbUtil.getConnection();

int newamt=0;

System.out.println(" Conneted in updateAmt.............");

try

{

con.setAutoCommit(false);

pst1=con.prepareStatement("select billamount from bills where mobilenumber=?");

pst1.setString(1, userDTOt.getMobilenumber());

**rs=pst1.executeQuery();**

rs.next();

int oldbillamt=rs.getInt(1);

System.out.println(" bill amount is selected successfully "+oldbillamt);

if(amt.intValue()<=0)

{

newamt=0;

}

else

{

newamt=oldbillamt-amt.intValue();

}

pst2=con.prepareStatement("update bills set billamount =? where mobilenumber=?");

pst2.setInt(1, newamt);

pst2.setString(2, userDTOt.getMobilenumber());

int noOfRec2=pst2.executeUpdate();

System.out.println(" New Amt is updated successfully in bilss table");

con.commit();

}

catch (SQLException e)

{

try

{

con.rollback();

}

catch (SQLException e1)

{

throw new UsersException("some exception in insert User method"+e1.getMessage() );

}

throw new UsersException("some exception in insert User method"+e.getMessage() );

}

finally

{

try

{

pst1.close();

pst2.close();

con.close();

}

catch (SQLException e)

{

throw new UsersException("some exception while closing objects "+e.getMessage() );

}

}

return newamt;

}

public ArrayList<String> getUsers()throws UsersException

{

con=DbUtil.getConnection();

ArrayList<String> userList=new ArrayList<String>();

System.out.println(" Conneted in getUsers.............");

try

{

con.setAutoCommit(false);

pst1=con.prepareStatement("select username from users");

rs=pst1.executeQuery();

while(rs.next())

{

userList.add(rs.getString(1));

}

con.commit();

}

catch (SQLException e)

{

try

{

con.rollback();

}

catch (SQLException e1)

{

throw new UsersException("some exception in insert User method"+e1.getMessage() );

}

throw new UsersException("some exception in insert User method"+e.getMessage() );

}

finally

{

try

{

pst1.close();

rs.close();

con.close();

}

catch (SQLException e)

{

throw new UsersException("some exception while closing objects "+e.getMessage() );

}

}

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

return userList;

}

public UserDto getBillAmt(String username)throws UsersException

{

UserDto userDto=new UserDto();

con=DbUtil.getConnection();

System.out.println(" Conneted in getBillAmt............."+username);

try

{

con.setAutoCommit(false);

pst1=con.prepareStatement("select \* from users where username=?");

pst1.setString(1, username);

rs=pst1.executeQuery();

System.out.println("Result set ............."+rs);

while(rs.next())

{

userDto.setName(rs.getString(1));

userDto.setUsername(rs.getString(2));

userDto.setPassword(rs.getString(3));

userDto.setMobilenumber(rs.getString(4));

}

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

pst2=con.prepareStatement("select \* from bills where mobilenumber=?");

pst2.setString(1, userDto.getMobilenumber());

rs2=pst2.executeQuery();

rs2.next();

userDto.setBillamount(rs2.getInt(2));

System.out.println(" user............. "+userDto);

con.commit();

}

catch (SQLException e)

{

try

{

con.rollback();

}

catch (SQLException e1)

{

throw new UsersException("some exception in getBillAmt method"+e1.getMessage() );

}

throw new UsersException("some exception in getBillAmt method"+e.getMessage() );

}

finally

{

try

{

pst1.close();

pst2.close();

rs2.close();

rs.close();

con.close();

}

catch (SQLException e)

{

throw new UsersException("some exception while closing objects "+e.getMessage() );

}

}

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

return userDto;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* UserDto.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**package** com.igate.dto;

**public** **class** UserDto

{

**private** String name;

**private** String username;

**private** String password;

**private** String mobilenumber;

**private** **double** billamount;

**public** String getMobilenumber() {**return** mobilenumber;}

**public** **void** setMobilenumber(String mobilenumber) { **this**.mobilenumber = mobilenumber;

}

**public** **double** getBillamount() { **return** billamount; }

**public** **void** setBillamount(**double** billamount) { **this**.billamount = billamount;}

**public** String getName() {**return** name; }

**public** **void** setName(String name) {**this**.name = name; }

**public** String getUsername() { **return** username; }

**public** **void** setUsername(String username) {**this**.username = username; }

**public** String getPassword() { **return** password; }

**public** **void** setPassword(String password) {**this**.password = password; }

@Override

**public** String toString() {

**return** "UserDto [name=" + name + ", username=" + username

+ ", password=" + password + ", mobilenumber=" + mobilenumber

+ ", billamount=" + billamount + "]";

}}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* UsersException.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.exception;

public class UsersException extends Exception {

public UsersException(String msg)

{super(msg); }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CompnameFilter.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.filter;

import java.io.IOException;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

public class CompnameFilter implements Filter {

public CompnameFilter() {

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

}

public void destroy() {

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

}

public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {

chain.doFilter(request, response);

}

public void init(FilterConfig fConfig) throws ServletException {

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

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* LoggingFilter .java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.filter;

import java.io.IOException;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import org.apache.log4j.Logger;

import org.apache.log4j.PropertyConfigurator;

import java.util.Date;

public class LoggingFilter implements Filter {

public LoggingFilter() {

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

}

public void destroy() {

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

}

public void doFilter(ServletRequest request, ServletResponse response,

FilterChain chain) throws IOException, ServletException {

System.out.println("Login Filtered doFilter......................");

String username = request.getParameter("uName");

Logger logger = Logger.getLogger(LoggingFilter.class);

PropertyConfigurator

.configure("myprop.properties");

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

logger.warn("Here is some user tried to login " + username + "on "

+ new Date());

logger.error("Here is some user tried to login " + username + "on "

+ new Date());

logger.fatal("Here is some user tried to login " + username + "on "

+ new Date());

chain.doFilter(request, response);

}

public void init(FilterConfig fConfig) throws ServletException {

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

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* UserService .java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.service;

import java.sql.Connection;

import java.sql.SQLException;

import java.util.ArrayList;

import javax.naming.NamingException;

import com.igate.dao.UserDao;

import com.igate.dao.UserDaoImpl;

import com.igate.dto.UserDto;

import com.igate.exception.UsersException;

import com.igate.util.DbUtil;

public class UserService

{

public void insertUsers(UserDto userDTO) throws UsersException

{

UserDao userDao=new UserDaoImpl();

userDao.insertUsers(userDTO);

}

public int updateAmt(Integer amt,UserDto userDTO)throws UsersException

{

UserDao userDao=new UserDaoImpl();

return userDao.updateAmt( amt, userDTO);

}

public ArrayList<String> getUsers()throws UsersException

{

UserDao userDao=new UserDaoImpl();

return userDao.getUsers();

}

public UserDto getBillAmt(String username)throws UsersException

{

UserDao userDao=new UserDaoImpl();

return userDao.getBillAmt(username);

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DbUtil .java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.util;

import java.sql.Connection;

import java.sql.SQLException;

import javax.naming.InitialContext;

import javax.naming.NamingException;

import javax.sql.DataSource;

import com.igate.exception.UsersException;

public class DbUtil

{

public static Connection getConnection()throws UsersException

{

InitialContext ic;

DataSource ds;

try

{

ic = new InitialContext();

ds=(DataSource)ic.lookup("java:/OracleDS");

// DataSource ds=(DataSource)ic.lookup("MyDS");

return ds.getConnection();

}

catch (Exception e)

{

throw new UsersException("some databse eror..."+e.getMessage());

}

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* myprop.properties\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

# Log levels

# Uncomment the following line to enable full loggin for every class

#log4j.rootLogger=trace, stdout, R

log4j.rootLogger=warn, R

#log4j.logger.pkg=trace, stdout, R

# Console appender configuration

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

# Pattern to output the caller's file name and line number.

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%5p [%t] (%F:%L) - %m%n

# Rolling File Appender

log4j.appender.R=org.apache.log4j.RollingFileAppender

# Path and file name to store the log file.

log4j.appender.R.File=D:/log/vaishali.log

#log4j.appender.R.File=C:\\JavaMptWS\\vaishali.log

log4j.appender.R.MaxFileSize=500KB

# Keep one backup file

log4j.appender.R.MaxBackupIndex=1

# Rolling File Appender layout

log4j.appender.R.layout=org.apache.log4j.PatternLayout

log4j.appender.R.layout.ConversionPattern=%d - %c - %p - %m%n

log4j.rootLogger=ALL, A2

########## Appender A2

log4j.appender.A2=org.apache.log4j.FileAppender

log4j.appender.A2.File=/MyMptWebP/bookinglog.txt

# Append to the end of the file or overwrites the file at start.

log4j.appender.A2.Append=true

log4j.appender.A2.layout=org.apache.log4j.SimpleLayout

################log4j.appender.A2.layout.ConversionPattern=[%5p] %d{mm:ss} (%F:%M:%L)%n%m%n%n

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*web.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"

id="WebApp\_ID" version="2.5">

<display-name>MyMptWebP</display-name>

<welcome-file-list>

<welcome-file>html/home.html</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>ProcessUser</servlet-name>

<servlet-class>com.igate.controller.ProcessUser</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ProcessUser</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

<filter>

<filter-name>LoggingFilter</filter-name>

<filter-class>com.igate.filter.LoggingFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>LoggingFilter</filter-name>

<url-pattern>\*.do</url-pattern>

</filter-mapping>

<filter>

<display-name>CompnameFilter</display-name>

<filter-name>CompnameFilter</filter-name>

<filter-class>com.igate.filter.CompnameFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>CompnameFilter</filter-name>

<url-pattern>\*.do</url-pattern>

</filter-mapping>

<security-constraint>

<web-resource-collection>

<web-resource-name>Sensitive</web-resource-name>

<url-pattern>/jsps/Register.jsp</url-pattern>

<!-- <url-pattern>/\*</url-pattern> -->

<http-method>GET</http-method>

<http-method>POST</http-method>

</web-resource-collection>

<auth-constraint>

<role-name>admin\_user</role-name>

</auth-constraint>

<user-data-constraint>

<transport-guarantee>CONFIDENTIAL</transport-guarantee>

</user-data-constraint>

</security-constraint>

<!-- All security roles that will be used anywhere within this app. -->

<security-role>

<role-name>admin\_user</role-name>

</security-role>

<!-- Tell the server to use BASIC authentication. -->

<login-config>

<auth-method>BASIC</auth-method>

<realm-name>Very Sensitive Application</realm-name>

</login-config>

</web-app>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*jboss-web.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version=*"1.0"* encoding=*"ISO-8859-1"*?>

<!DOCTYPE jboss-web

PUBLIC "-//JBoss//DTD Web Application 2.3V2//EN"

"http://www.jboss.org/j2ee/dtd/jboss-web\_3\_2.dtd">

<jboss-web>

<!-- A security domain that restricts access-->

<security-domain>java:/jaas/MyMptWebP</security-domain>

<!-- <context-root>jbossws</context-root> -->

</jboss-web>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*validation.js\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**function** validation() {

**var** custname = document.RegisterUserForm.custName.value;

**var** mobileNo = document.RegisterUserForm.mobileNo.value;

**var** uname = document.RegisterUserForm.uName.value;

**var** pass = document.RegisterUserForm.pass.value;

**var** repass = document.RegisterUserForm.rePass.value;

**var** upattern = /[a-zA-Z]/;

**var** number = /[0-9]{10}/;

**if** (custname == "" || custname == **null**) {

**var** custNameId=document.getElementById("custNameId");

custNameId.innerHTML="Please enter your name";

**return** **false**;

}

**if** (!custname.match(upattern)) {

//alert("Name should be in characters only");

**var** custNameId=document.getElementById("custNameId");

custNameId.innerHTML="Name should be in characters only";

**return** **false**;

}

**if** (mobileNo == "" || mobileNo == **null**) {

//alert("Please enter your Mobile number");

**var** mobileNoId=document.getElementById("mobileNoId");

mobileNoId.innerHTML="Please enter your Mobile number";

**return** **false**;

}

**if** (!mobileNo.match(number)) {

//alert("Mobile number in digits only and maximum length is 10");

**var** mobileNoId=document.getElementById("mobileNoId");

mobileNoId.innerHTML="Mobile number in digits only and maximum length is 10";

**return** **false**;

}

**if** (uname == "" || uname == **null**) {

//alert("Please enter your Username");

**var** uNameId=document.getElementById("uNameId");

uNameId.innerHTML="Please enter your Username";

**return** **false**;

}

**if** (pass == "" || pass == **null**)

{

//alert("Please enter your Password");

**var** passId=document.getElementById("passId");

passId.innerHTML="Please enter your Password";

**return** **false**;

}

**if** (repass == "" || repass == **null**)

{

//alert("Please re-enter your Password")

**var** rePassId=document.getElementById("rePassId");

rePassId.innerHTML="Please re-enter your Password";

**return** **false**;

}

**if** (repass != pass)

{

//alert("Password didnt match")

**var** rePassId=document.getElementById("rePassId");

rePassId.innerHTML="Password didnt match";

**return** **false**;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*home.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body bgcolor=*"pink"*>

<h4>Welcome To Mobile Billing System.....</h4>

<a href=*"ProcessUser.do?action=Login"*>Login</a><br/>

<a href=*"jsps/Register.jsp"*>Register</a>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*error.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"* isErrorPage=*"true"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<%= "..............Error"+exception.getMessage() %>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*customerHome.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@taglib uri=*"http://java.sun.com/jstl/core\_rt"* prefix=*"c"* %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body bgcolor=*"pink"*>

<center>

<h2>AirSpace Telecomm</h2>

<br>

<h3>Welcome ${UserDetails.name}</h3>

<br>

<br>

<form action=*"ProcessUser.do"*>

<h4><pre>The pending bill amount for your

Mobile Number : ${UserDetails.mobilenumber}

is : Rs. ${UserDetails.billamount}</pre></h4><br><br>

<input type=*"submit"* name=*"action"* value =*"PayBill"*>

</form>

</center>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*login.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@taglib uri=*"http://java.sun.com/jstl/core\_rt"* prefix=*"c"* %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<center>

<h3> Enter UR Login Details....</h3>

<c:out value=*"*${requestScope.UserNames}*"*></c:out>

<form name=*"LoginUserForm"* action=*"ProcessUser.do"* method=*"get"*

onSubmit="return validation();">

<table>

<tr>

<td>Usernames :</td>

<td>

<select name=*"uName"* >

<c:forEach items=*"*${UserNames}*"* var =*"nm"*>

<option value=*'*<c:out value="${nm}"></c:out>*'*><c:out value=*"*${nm}*"*></c:out></option>

</c:forEach>

</select>

</td>

</tr>

<tr>

<td>Password :</td>

<td><input type=*"password"* name=*"pass"* /></td>

</tr>

<tr>

<td></td>

<td><input type=*"submit"* name=*"action"*

value=*"ShowBillAmt"* ></td>

</tr>

</table>

</form>

</center>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*payBills.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@taglib uri=*"http://java.sun.com/jstl/core\_rt"* prefix=*"c"* %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body bgcolor=*"pink"*>

<center>

<h2>AirSpace Telecomm</h2>

<br>

<h3>Pay your Bill Here</h3>

<br> <br>

<h3>${UserDetails.name}</h3>

<h4> Total Payment for the Mobile number : ${UserDetails.mobilenumber}</h4>

<h4> IS : Rs. ${UserDetails.billamount}</h4>

</br>

<c:set var=*"amt"* value=*"*${AMOUNT}*"* />

Amount U Want To Pay Is: <c:out value=*"*${amt}*"* />

<c:set var=*"actualamt"* value=*"*${UserDetails.billamount}*"* />

<form action=*"ProcessUser.do"*>

<table>

<tr>

<td>Enter the amount to Pay</td>

<td><input type=*"text"* name=*"amount"* />

</td>

</tr>

<tr>

<td>Enter the 10 digit card number</td>

<td><input type=*"text"* name=*"cardnumber"* />

</td>

</tr>

</table>

<input type=*"submit"* name=*"action"* value=*"Pay"*>

</form>

<c:if test=*"*${(amt **gt** actualamt) **or**(amt <= 0)}*"*>

<c:out value=*"Sorry Amt to pay is more than actula Amount Or Negative"* /> <br/>

</c:if>

<c:if test=*"*${(amt <= actualamt)&&(amt > 0)}*"*>

<jsp:forward page=*"../ProcessUser.do"*>

<jsp:param value=*"Success"* name=*"action"*/>

</jsp:forward>

</c:if>

<%-- <c:if test="${amt <= 0}">

<c:out value="Sorry Amt Is Negative or zero" /> <br/>

</c:if> --%>

</center>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Register.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"* isErrorPage=*"false"* errorPage=*"errors.jsp"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

<script type=*"text/javascript"* language=*"javascript"* src=*"../js/validation.js"*>

</script>

</head>

<body bgcolor=*"skyblue"*>

<center>

<h2>AirSpace Telecomm</h2>

<br>

<h3>New User Registration Form</h3>

<br> <br>

<form name=*"RegisterUserForm"* action=*"ProcessUser.do"* method=*"get"* onSubmit="return validation();">

<table>

<tr>

<td>Enter your Name :</td>

<td><input type=*"text"* name=*"custName"* /></td>

<td ><div id=*"custNameId"*></div></td>

</tr>

<tr>

<td>Mobile Number :</td>

<td><input type=*"text"* name=*"mobileNo"* /></td>

<td ><div id=*"mobileNoId"*></div></td>

</tr>

<tr>

<td>Username :</td>

<td><input type=*"text"* name=*"uName"* /></td>

<td ><div id=*"uNameId"*></div></td>

</tr>

<tr>

<td>Password :</td>

<td><input type=*"text"* name=*"pass"* /></td>

<td ><div id=*"passId"*></div></td>

</tr>

<tr>

<td>Re-enter Password :</td>

<td><input type=*"text"* name=*"rePass"* /></td>

<td ><div id=*"rePassId"*></div></td>

</tr>

<tr>

<td></td>

<td><input type=*"submit"* name=*"action"*

value=*"Register"* ></td>

</tr>

</table>

</form>

</center>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*success.jsp\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@taglib uri=*"http://java.sun.com/jstl/core\_rt"* prefix=*"c"* %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body bgcolor=*"cyan"*>

Welcome ur pending bill is : <c:out value=*"*${requestScope.PendingAmt}*"* />

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Steps\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DB\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

create table users(name varchar2(40),username varchar2(40) primary key,

password varchar2(15),mobilenumber varchar2(12));

C:\MyInstallations\Jboss4\jboss-4.2.3.GA\server\default\conf

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*jbossws-users.properties\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# A sample users.properties file for use with the #UsersRolesLoginModule

vaishali = vaishali123

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*jbossws-roles.properties\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

# A sample roles.properties file for use with the #UsersRolesLoginModule

#vaishali=role

vaishali = admin\_user

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*login-config.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*8**

<?xml version='1.0'?>

<!DOCTYPE policy PUBLIC

"-//JBoss//DTD JBOSS Security Config 3.0//EN"

"http://www.jboss.org/j2ee/dtd/security\_config.dtd">

<!-- The XML based JAAS login configuration read by the

org.jboss.security.auth.login.XMLLoginConfig mbean. Add

an application-policy element for each security domain.

The outline of the application-policy is:

<application-policy name="security-domain-name">

<authentication>

<login-module code="login.module1.class.name" flag="control\_flag">

<module-option name = "option1-name">option1-value</module-option>

<module-option name = "option2-name">option2-value</module-option>

...

</login-module>

<login-module code="login.module2.class.name" flag="control\_flag">

...

</login-module>

...

</authentication>

</application-policy>

$Revision: 64598 $

-->

<policy>

<!-- Used by clients within the application server VM such as

mbeans and servlets that access EJBs.

-->

<application-policy name = "client-login">

<authentication>

<login-module code = "org.jboss.security.ClientLoginModule"

flag = "required">

<!-- Any existing security context will be restored on logout -->

<module-option name="restore-login-identity">true</module-option>

</login-module>

</authentication>

</application-policy>

<!-- Security domain for JBossMQ -->

<application-policy name = "jbossmq">

<authentication>

<login-module code = "org.jboss.security.auth.spi.DatabaseServerLoginModule"

flag = "required">

<module-option name = "unauthenticatedIdentity">guest</module-option>

<module-option name = "dsJndiName">java:/DefaultDS</module-option>

<module-option name = "principalsQuery">SELECT PASSWD FROM JMS\_USERS WHERE USERID=?</module-option>

<module-option name = "rolesQuery">SELECT ROLEID, 'Roles' FROM JMS\_ROLES WHERE USERID=?</module-option>

</login-module>

</authentication>

</application-policy>

<!-- Security domain for JBossMQ when using file-state-service.xml

<application-policy name = "jbossmq">

<authentication>

<login-module code = "org.jboss.mq.sm.file.DynamicLoginModule"

flag = "required">

<module-option name = "unauthenticatedIdentity">guest</module-option>

<module-option name = "sm.objectname">jboss.mq:service=StateManager</module-option>

</login-module>

</authentication>

</application-policy>

-->

<!-- Security domains for testing new jca framework -->

<application-policy name = "HsqlDbRealm">

<authentication>

<login-module code = "org.jboss.resource.security.ConfiguredIdentityLoginModule"

flag = "required">

<module-option name = "principal">sa</module-option>

<module-option name = "userName">sa</module-option>

<module-option name = "password"></module-option>

<module-option name = "managedConnectionFactoryName">jboss.jca:service=LocalTxCM,name=DefaultDS</module-option>

</login-module>

</authentication>

</application-policy>

<application-policy name = "JmsXARealm">

<authentication>

<login-module code = "org.jboss.resource.security.ConfiguredIdentityLoginModule"

flag = "required">

<module-option name = "principal">guest</module-option>

<module-option name = "userName">guest</module-option>

<module-option name = "password">guest</module-option>

<module-option name = "managedConnectionFactoryName">jboss.jca:service=TxCM,name=JmsXA</module-option>

</login-module>

</authentication>

</application-policy>

<!-- A template configuration for the jmx-console web application. This

defaults to the UsersRolesLoginModule the same as other and should be

changed to a stronger authentication mechanism as required.

-->

<application-policy name = "jmx-console">

<authentication>

<login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule"

flag = "required">

<module-option name="usersProperties">props/jmx-console-users.properties</module-option>

<module-option name="rolesProperties">props/jmx-console-roles.properties</module-option>

</login-module>

</authentication>

</application-policy>

<!-- A template configuration for the web-console web application. This

defaults to the UsersRolesLoginModule the same as other and should be

changed to a stronger authentication mechanism as required.

-->

<application-policy name = "web-console">

<authentication>

<login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule"

flag = "required">

<module-option name="usersProperties">web-console-users.properties</module-option>

<module-option name="rolesProperties">web-console-roles.properties</module-option>

</login-module>

</authentication>

</application-policy>

<!--

A template configuration for the JBossWS security domain.

This defaults to the UsersRolesLoginModule the same as other and should be

changed to a stronger authentication mechanism as required.

-->

<application-policy name="JBossWS">

<authentication>

<login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule"

flag="required">

<module-option name="usersProperties">props/jbossws-users.properties</module-option>

<module-option name="rolesProperties">props/jbossws-roles.properties</module-option>

<module-option name="unauthenticatedIdentity">anonymous</module-option>

</login-module>

</authentication>

</application-policy>

<application-policy name="MyMptWebP">

<authentication>

<login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule"

flag="required">

<module-option name="usersProperties">props/jbossws-users.properties</module-option>

<module-option name="rolesProperties">props/jbossws-roles.properties</module-option>

</login-module>

</authentication>

</application-policy>

<application-policy name="TestApp">

<authentication>

<login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule"

flag="required">

<module-option name="usersProperties">props/jbossws-users.properties</module-option>

<module-option name="rolesProperties">props/jbossws-roles.properties</module-option>

</login-module>

</authentication>

</application-policy>

<!-- The default login configuration used by any security domain that

does not have a application-policy entry with a matching name

-->

<application-policy name = "other">

<!-- A simple server login module, which can be used when the number

of users is relatively small. It uses two properties files:

users.properties, which holds users (key) and their password (value).

roles.properties, which holds users (key) and a comma-separated list of

their roles (value).

The unauthenticatedIdentity property defines the name of the principal

that will be used when a null username and password are presented as is

the case for an unuathenticated web client or MDB. If you want to

allow such users to be authenticated add the property, e.g.,

unauthenticatedIdentity="nobody"

-->

<authentication>

<login-module code = "org.jboss.security.auth.spi.UsersRolesLoginModule"

flag = "required" />

</authentication>

</application-policy>

</policy>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Jboss-web.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE jboss-web

PUBLIC "-//JBoss//DTD Web Application 2.3V2//EN"

"http://www.jboss.org/j2ee/dtd/jboss-web\_3\_2.dtd">

<jboss-web>

<!-- A security domain that restricts access-->

<security-domain>java:/jaas/MyMptWebP</security-domain>

<!-- <context-root>jbossws</context-root> -->

</jboss-web>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SSL Security\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Step-1)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C:\>keytool -genkey -alias jbkeystore -keypass jbadmin -keystore C:\jbkeystore.bin -storepass jbadmin

What is your first and last name?

[Unknown]: Vaishali S

What is the name of your organizational unit?

[Unknown]: ilearn

What is the name of your organization?

[Unknown]: igate

What is the name of your City or Locality?

[Unknown]: pune

What is the name of your State or Province?

[Unknown]: ms

What is the two-letter country code for this unit?

[Unknown]: in

Is CN=Vaishali S, OU=ilearn, O=igate, L=pune, ST=ms, C=in correct?

[no]: yes

**Step-2) Copy above bin file in the following folder**

C:\MyInstallations\Jboss4\jboss-4.2.3.GA\server\default\conf

**Step-4) Edit Server.xml from**

C:\MyInstallations\Jboss4\jboss-4.2.3.GA\server\default\deploy\jboss-web.deployer

<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"

maxThreads="150" scheme="https" secure="true"

clientAuth="false"

keystoreFile="${jboss.server.home.dir}/conf/jbkeystore.bin"

keystorePass="jbadmin" sslProtocol="TLS" />

**Step-5) Start the server**

https://localhost:8443/

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***