**作业报告6**

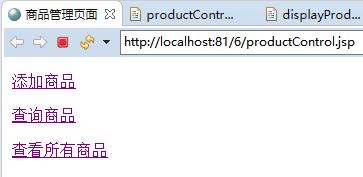
1、测试查询商品信息的例子,使之能正常运行，并增加如下功能。

（1）分别增加添加、修改和删除功能；

（2）增加一个商品管理页面，包含添加商品、查询商品、查看所有商品等功能，查询结果列表的后面增加修改和删除功能。

**1.1运行结果**

(1)商品管理页面



(2)添加商品页面



(3)添加商品信息结果页面



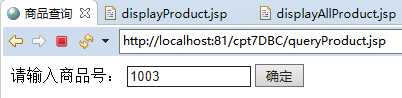
添加商品前：



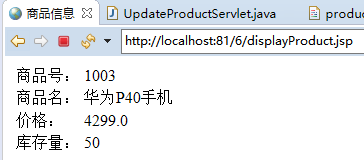
添加商品后：

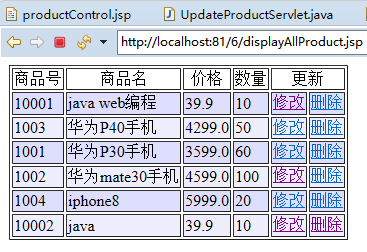


(4)查询商品页面

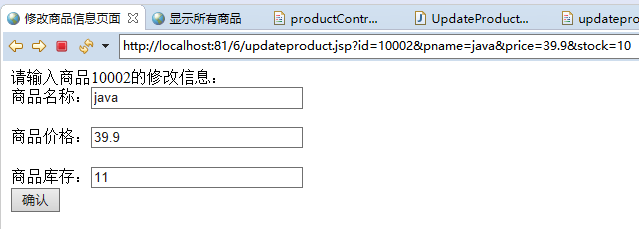


(5)查询商品结果页面



(6)显示所有商品页面

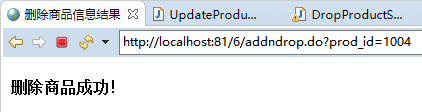
(7)修改商品信息页面



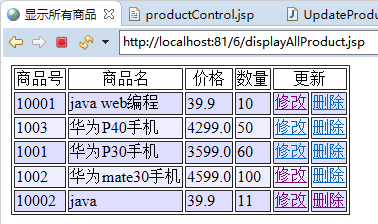
(8)修改商品信息结果页面



(9)删除商品信息结果页面



(10)修改删除操作后：



**1.2体会**

1. 超链接跳转到servlet使用的是doGet()方法.表单提交到doPost()方法中.
2. 超链接传递中文参数出现了中文乱码问题，我在解码参数前加了request.setCharacterEncoding("utf-8");并试了多种方法都没用.
3. 在request.getParameter()时要注意类型转换，PreparedStatement.setXxx()也要注意设置参数的类型.
4. 修改信息时，不修改商品编号，可修改的另外3个参数可以先在输入框中显示原始值(具有初始value)，这样若有未修改的，则把原始值传给servlet，方便把3个参数一起更新.
5. 添加商品提交表单交给doPost()处理，删除商品用超链接跳转交给doGet()处理，则可以把2个功能合并到一个AddnDropProductServlet.java中处理.

**1.3源代码**(只放上处理查询、添加和删除、修改功能的servlet)

1. **QueryProductServlet.java**

@WebServlet("/queryproduct.do")

**public** **class** QueryProductServlet **extends** HttpServlet {

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

Connection dbconn = **null**;

**public** **void** init() {

String driver = "org.postgresql.Driver";

String dburl = "jdbc:postgresql://127.0.0.1:5432/postgres";

String username = "postgres";

String password = "6666";

**try** {

Class.*forName*(driver);

dbconn = DriverManager.*getConnection*(dburl, username, password);

} **catch** (ClassNotFoundException e1) {

System.***out***.println(e1);

} **catch** (SQLException e2) {

}

}

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

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

**try** {

String sql = "SELECT \* FROM products WHERE prod\_id = ?";

PreparedStatement pstmt = dbconn.prepareStatement(sql);

pstmt.setString(1, productid);

ResultSet rst = pstmt.executeQuery();

**if** (rst.next()) {

Product product = **new** Product();

product.setProd\_id(rst.getString("prod\_id"));

product.setPname(rst.getString("pname"));

product.setPrice(rst.getDouble("price"));

product.setStock(rst.getInt("stock"));

request.getSession().setAttribute("product", product);

response.sendRedirect("displayProduct.jsp");

} **else** {

response.sendRedirect("error.jsp");

}

} **catch** (SQLException e) {

e.printStackTrace();

}

}

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

ArrayList<Product> productList = **null**;

productList = **new** ArrayList<Product>();

**try** {

String sql = "SELECT \* FROM products";

PreparedStatement pstmt = dbconn.prepareStatement(sql);

ResultSet rst= pstmt.executeQuery();

**while** (rst.next()) {

Product product = **new** Product();

product.setProd\_id(rst.getString("prod\_id"));

product.setPname(rst.getString("pname"));

product.setPrice(rst.getDouble("price"));

product.setStock(rst.getInt("stock"));

productList.add(product);

}

**if** (!productList.isEmpty()) {

request.getSession().setAttribute("productList", productList);

response.sendRedirect("displayAllProduct.jsp");

} **else** {

response.sendRedirect("error.jsp");

}

} **catch** (SQLException e) {

e.printStackTrace();

}

}

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

**try** {

dbconn.close();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

1. **AddnDropProductServlet.java**

@WebServlet("/addndrop.do")

**public** **class** AddnDropProductServlet **extends** HttpServlet {

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

Connection dbconn = **null**;

**public** **void** init() {//getConnection

String driver = "org.postgresql.Driver";

String dburl = "jdbc:postgresql://127.0.0.1:5432/postgres";

String username = "postgres";

String password = "6666";//用自己设置的密码

**try** {

Class.*forName*(driver);

dbconn = DriverManager.*getConnection*(dburl, username, password);

} **catch** (ClassNotFoundException e1) {

System.***out***.println(e1);

} **catch** (SQLException e2) {

}

}

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

//添加商品提交表单->doPost

request.setCharacterEncoding("utf-8");

String message = **null**;

Product product = **new** Product();

String prod\_id = request.getParameter("prod\_id");

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

**double** price = Double.*parseDouble*(request.getParameter("price"));

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

**try** {

String sql = "INSERT INTO products VALUES (?, ?, ?, ?) ";

PreparedStatement pstmt = dbconn.prepareStatement(sql);

pstmt.setString(1, prod\_id);

pstmt.setString(2, pname);

pstmt.setDouble(3, price);

pstmt.setInt(4, stock);

**int** n = pstmt.executeUpdate();

product.setProd\_id(prod\_id);

product.setPname(pname);

product.setPrice(price);

product.setStock(stock);

**if** (n>0) {

message = "<li>添加商品成功！</li>";

} **else** {

message = "<li>添加商品失败！</li>"; }

} **catch** (SQLException e) {

e.printStackTrace();

message = "<li>添加商品异常！</li>";

}

request.setAttribute("result", message);

request.setAttribute("product", product);

RequestDispatcher rd = getServletContext().getRequestDispatcher("/addproductresult.jsp");

rd.forward(request, response);

}

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

//超链接传递参数->doGet

request.setCharacterEncoding("utf-8");

String message = **null**;

String prod\_id = request.getParameter("id");

**try** {

String sql = "DELETE FROM products WHERE prod\_id=?";

PreparedStatement pstmt = dbconn.prepareStatement(sql);

pstmt.setString(1, prod\_id);

**int** n = pstmt.executeUpdate();

**if** (n>0) {

message = "<li>删除商品成功！</li>";

} **else** {

message = "<li>删除商品失败！</li>"; }

} **catch** (SQLException e) {

e.printStackTrace();

message = "<li>删除商品异常！</li>";

}

request.setAttribute("result", message);

RequestDispatcher rd = getServletContext().getRequestDispatcher("/dropproductresult.jsp");

rd.forward(request, response);

}

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

**try** {dbconn.close();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

1. **UpdateProductServlet.java**

@WebServlet("/updateproduct.do")

**public** **class** UpdateProductServlet **extends** HttpServlet {

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

Connection dbconn = **null**;

**public** **void** init() {//getConnection

String driver = "org.postgresql.Driver";

String dburl = "jdbc:postgresql://127.0.0.1:5432/postgres";

String username = "postgres";

String password = "6666";//用自己设置的密码

**try** {

Class.*forName*(driver);

dbconn = DriverManager.*getConnection*(dburl, username, password);

} **catch** (ClassNotFoundException e1) {

System.***out***.println(e1);

} **catch** (SQLException e2) {

}

}

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

request.setCharacterEncoding("utf-8");

String message = **null**;

Product product = **new** Product();

String prod\_id = request.getParameter("id");

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

**double** price = Double.*parseDouble*(request.getParameter("price"));

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

**try** {

String sql = "UPDATE products SET pname=?,price=?,stock=? WHERE prod\_id =?";

PreparedStatement pstmt = dbconn.prepareStatement(sql);

pstmt.setString(1, pname);

pstmt.setDouble(2, price);

pstmt.setInt(3, stock);

pstmt.setString(4, prod\_id);

**int** n = pstmt.executeUpdate();

product.setProd\_id(prod\_id);

product.setPname(pname);

product.setPrice(price);

product.setStock(stock);

**if** (n>0) {

message = "<li>更新商品成功！</li>";

} **else** {

message = "<li>更新商品失败！</li>"; }

} **catch** (SQLException e) {

e.printStackTrace();

message = "<li>更新商品异常！</li>";

}

request.setAttribute("result", message);

request.setAttribute("product", product);

RequestDispatcher rd = getServletContext().getRequestDispatcher("/updateproductresult.jsp");

rd.forward(request, response);

}

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

**try** {

dbconn.close();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

2、实现某师生健康码管理系统的用户注册和登录功能，具体要求如下：

（1）用户注册register.jsp。如果是教师，则可输入姓名、身份证号、工号、密码、学院、角色（系统管理员、校级管理员、院级管理员、普通教师）等信息，注册的信息写入教师表dbt\_teacher中；如果是学生，则可输入姓名、身份证号、学号、密码、学院、专业、班级等信息，注册的信息存入学生表dbt\_student。要求密码采用SHA256加密（可用JDK自带的java.security.MessageDigest实现）后存入数据库。要求教师的身份证号、工号和学生的身份证号、学号不能重复注册，否则跳转至failed.jsp提示“您输入的身份证号、工号或学号已被注册！”。

（2）用户登录login.jsp。选择用户类型(教师或学生)，输入登录名、密码，教师的登录名为工号，学生的登录名为学号。如果教师用户登录成功后，跳转至teacher.jsp显示该教师的姓名、身份证号、工号、密码、学院、角色等信息；如果是学生用户登录成功后，跳转至student.jsp显示该学生的姓名、身份证号、学号、密码、学院、专业、班级等信息；如果登录失败，则跳转至error.jsp，显示“您的用户名或密码错误”，3秒后自动跳转至登录页面login.jsp。

**2.1需求分析**

**1、功能描述**

输入注册信息可录入数据库，输入登录信息可显示用户信息。

**2、业务处理流程及要求**

（1）输入的信息不能为空，否则弹出提示框；不为空时，点击提交按钮处理信息；

（2）注册的信息存入数据库，密码采用SHA256加密，注册成功后跳转显示注册结果；身份证号、工号/学号不能重复注册，否则跳转至failed.jsp提示“您输入的身份证号、工号或学号已被注册！”；

（3）用户登录login.jsp。选择用户类型(教师或学生)，输入登录名、密码，教师的登录名为工号，学生的登录名为学号。如果登录成功，则跳转至用户信息页面显示登录结果和用户信息；如果登录失败，则跳转至error.jsp，显示“您的用户名或密码错误”，3秒后自动跳转至登录页面login.jsp。

**3、输入信息**

如果是教师，则可输入姓名、身份证号、工号、密码、学院、角色（系统管理员、校级管理员、院级管理员、普通教师）等信息；如果是学生，则可输入姓名、身份证号、学号、密码、学院、专业、班级等信息。

**4、输出信息**

注册后跳转显示注册结果；登录不成功则跳转显示登录结果；登录成功后跳转显示登录结果和用户信息。

**2.2开发步骤**

分析功能，有用户注册和登录2大块。

**(1)编写JavaBean对象:**创建User对象，包含姓名、身份证号、工号/学号、密码、学院5个基本属性,创建Teacher和Student对象继承User，Teacher增加角色属性、Student增加专业、班级属性；创建Sha256对象，用它的getSHA256(String str)方法实现密码加密。

**(2)注册的jsp：**

注册页面：表单中设置用户类型、姓名、身份证号、工号/学号、密码、学院6行基本输入框，用户类型下拉框通过onchange属性调用change()函数，当内容为“学生”时，7、8行显示专业、班级输入框，当内容为“教师”时，第7行显示“角色”下拉框，最后1行为“注册”、“取消”按钮。增加表单验证的js，输入框不能为空。

注册结果页面：第1行显示注册结果，第2行有“返回”和“登录”2个超链接。

重复注册页面：提示“您输入的身份证号、工号或学号已被注册！”，加上“请重新注册”的超链接。

**(3)RegisterServlet：**处理输入的注册信息，与数据库连接，若没有重复注册则存入用户信息。

**(4)登录的jsp：**

登录页面：表单中设置用户类型(教师或学生)下拉框，登录名(用户名)、密码输入框，“登录”和“取消”按钮，教师的登录名为工号，学生的登录名为学号。增加表单验证的js，输入框不能为空。

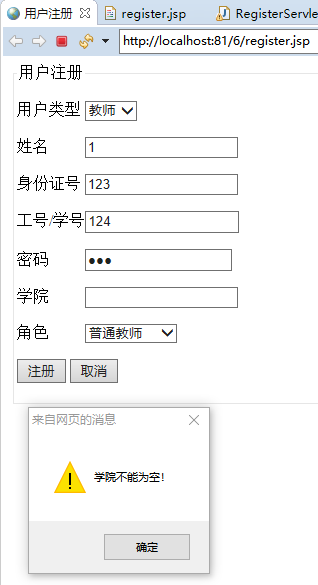
登录失败页面error.jsp：第1行显示“登陆失败！”的结果，第2行显示“您的用户名或密码错误”，第3行显示“3秒后自动跳转至登录页面”并自动递减秒数至1，第4行“显示若没有跳转,请按这里”，“这里”为登录页面的超链接。

teacher.jsp和student.jsp：第1行显示“登录成功！”的结果，然后以表格形式输出用户信息。

**(5)LoginServlet：**处理输入的登录信息，与数据库连接，匹配到则登录成功，把匹配到的用户信息赋给对应JavaBean对象，将对象添加到会话作用域，转发到对应用户类型页面；匹配不到则登录失败，跳转至error.jsp。

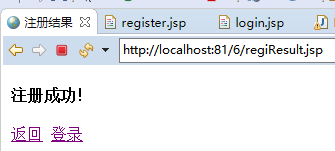
**2.3运行结果**

1. **注册页面**

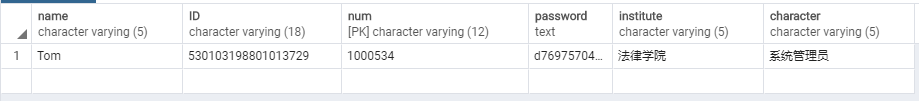
 

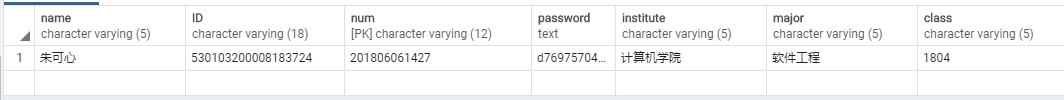


1. **注册结果页面**

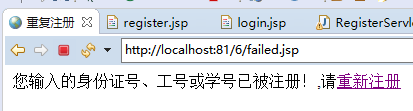


导入数据库：

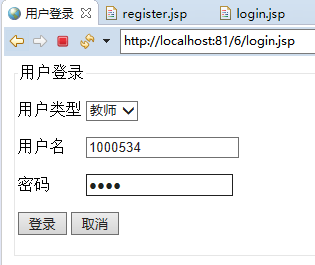
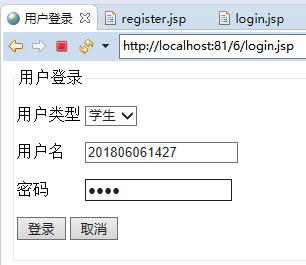




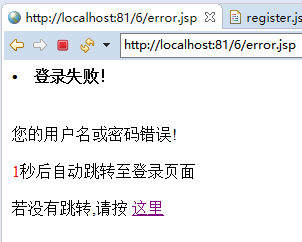
1. **重复注册页面**



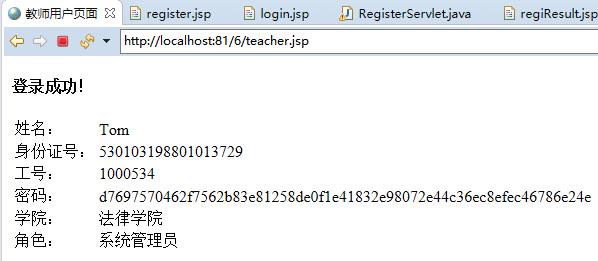
1. **登录页面**

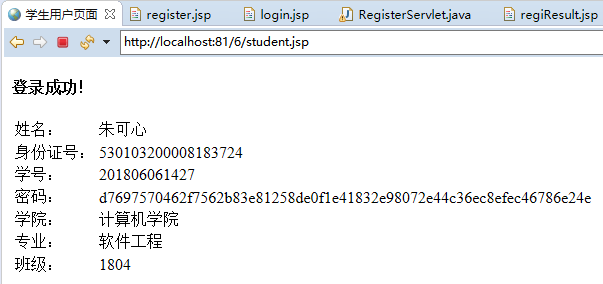
 

1. **登录失败页面**



1. **用户信息页面**

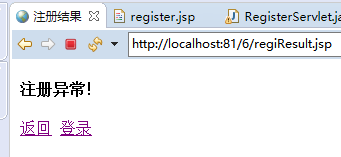




**2.4问题及解决方案**

1.原来在pgsql里设置密码属性类型为character(6)或character varying(6)都会出现报错，改成text就好了。

IMG_256



2.在RegisterServlet中检查重复注册时，跳转到failed.jsp那里会报错:

Cannot call sendRedirect() after the response has been committed.

意思是不能在一个response做了提交之后再重定向，也就是说再重定向之前不要去改变response，包括①修改response中参数的值，cookie,session等②重定向.

一般从2个方面解决问题：

1. 删除重写的doGet()、doPost()这两个方法里面的super.doGet()和super.doPost()语句。因为如果使用eclipse开发会自动生成重写方法的话，默认会去调用父类的doGet()、doPost()方法，方法中已经掉用过resp.sendError(...)了。
2. 在跳转之后加上return语句，因为跳转后，下面的代码已经完全无用，只会产生影响，所以加上return语句可以防止产生错误。

于是我写成这样就好了response.sendRedirect("failed.jsp"); **return** ;

**2.5实验收获**

通过本次实验我学会了将web应用和数据库结合起来，用eclipse连接数据库，将信息从前端页面输入，存入数据库后台，也可以将数据库中的信息导出到前端页面显示。核心内容如下。

数据库连接步骤：

（1）加载驱动程序。

（2）建立连接对象。

（3）创建语句对象。

（4）执行语句获得结果。

（5）关闭建立的对象。

测试访问数据库的示例：创建数据表，编写javabeans类，编写查询jsp页面，编写处理查询的servlet，编写结果显示jsp页面。

**2.6源代码**(只放上处理注册、登录用户信息的servlet)

1. **RegisterServlet.java**

@WebServlet("/register.do")

**public** **class** RegisterServlet **extends** HttpServlet {

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

Connection dbconn = **null**;

**public** **void** init() {//getConnection

String driver = "org.postgresql.Driver";

String dburl = "jdbc:postgresql://127.0.0.1:5432/postgres";

String username = "postgres";

String password = "6666";//用自己设置的密码

**try** {

Class.*forName*(driver);

dbconn = DriverManager.*getConnection*(dburl, username, password);

} **catch** (ClassNotFoundException e1) {

System.***out***.println(e1);

} **catch** (SQLException e2) {

}

}

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

request.setCharacterEncoding("utf-8");

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

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

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

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

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

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

String psw=Sha256.*getSHA256*(password);//加密

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

String message = **null**;

**if**(type.equals("教师")) {

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

**try** {//添加注册信息时ID、num不能重复

String sql = "SELECT \* FROM dbt\_teacher";

PreparedStatement pstmt = dbconn.prepareStatement(sql);

ResultSet rst= pstmt.executeQuery();

**while** (rst.next()) {

**if**(ID.equals(rst.getString("ID"))||num.equals(rst.getString("num"))) {

response.sendRedirect("failed.jsp"); **return** ;

}

}//String ==比较地址 equals()比较内容

//未重复则写入DB

sql = "INSERT INTO dbt\_teacher VALUES (?,?,?,?,?,?) ";

pstmt = dbconn.prepareStatement(sql);

pstmt.setString(1, name);

pstmt.setString(2, ID);

pstmt.setString(3, num);

pstmt.setString(4, psw);

pstmt.setString(5, institute);

pstmt.setString(6, character);

**int** n = pstmt.executeUpdate();

**if** (n>0) {

message = "<li>注册成功！</li>";

} **else** {

message = "<li>注册失败！</li>"; }

} **catch** (SQLException e) {

e.printStackTrace();

message = "<li>注册异常！</li>";

}

Teacher teacher=**new** Teacher(name,ID,num,psw,institute,character);

request.getSession().setAttribute("result", message);//直接用请求作用域跳转不了

request.getSession().setAttribute("teacher",teacher);

response.sendRedirect("regiResult.jsp");

}**else** {

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

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

**try** {

String sql = "SELECT \* FROM dbt\_student";

PreparedStatement pstmt = dbconn.prepareStatement(sql);

ResultSet rst= pstmt.executeQuery();

**while** (rst.next()) {

**if**(ID.equals(rst.getString("ID"))||num.equals(rst.getString("num"))) {

response.sendRedirect("failed.jsp"); **return** ;

}

}

//未重复则写入DB

sql = "INSERT INTO dbt\_student VALUES (?,?,?,?,?,?,?) ";

pstmt = dbconn.prepareStatement(sql);

pstmt.setString(1, name);

pstmt.setString(2, ID);

pstmt.setString(3, num);

pstmt.setString(4, psw);

pstmt.setString(5, institute);

pstmt.setString(6, major);

pstmt.setString(7, clas);

**int** n = pstmt.executeUpdate();

**if** (n>0) {

message = "<li>注册成功！</li>";

} **else** {

message = "<li>注册失败！</li>"; }

} **catch** (SQLException e) {

e.printStackTrace();

message = "<li>注册异常！</li>";

}

Student student=**new** Student(name,ID,num,psw,institute,major,clas);

request.getSession().setAttribute("result", message);

request.getSession().setAttribute("student",student);

response.sendRedirect("regiResult.jsp");

}

}

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

**try** {

dbconn.close();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

1. **LoginServlet.java**

@WebServlet("/login.do")

**public** **class** LoginServlet **extends** HttpServlet {

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

Connection dbconn = **null**;

**public** **void** init() {//getConnection

String driver = "org.postgresql.Driver";

String dburl = "jdbc:postgresql://127.0.0.1:5432/postgres";

String username = "postgres";

String password = "6666";//用自己设置的密码

**try** {

Class.*forName*(driver);

dbconn = DriverManager.*getConnection*(dburl, username, password);

} **catch** (ClassNotFoundException e1) {

System.***out***.println(e1);

} **catch** (SQLException e2) {

}

}

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

request.setCharacterEncoding("utf-8");

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

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

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

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

String psw=Sha256.*getSHA256*(password);//加密

String message = **null**;

**int** flag=0;

**if**(type.equals("教师")) {

**try** {

String sql = "SELECT \* FROM dbt\_teacher";

PreparedStatement pstmt = dbconn.prepareStatement(sql);

ResultSet rst= pstmt.executeQuery();

**while** (rst.next()) {

**if**(num.equals(rst.getString("num"))&&psw.equals(rst.getString("password"))) {

flag=1;

message = "<li>登录成功！</li>";

Teacher teacher=**new** Teacher(rst.getString("name"),rst.getString("ID"),num,psw,rst.getString("institute"),rst.getString("character"));

request.getSession().setAttribute("teacher",teacher);

**break**;

}

}

**if**(flag==0) {

message = "<li>登录失败！</li>";

}

}**catch** (SQLException e) {

e.printStackTrace();

message = "<li>登录异常！</li>";

}

request.getSession().setAttribute("result", message);

**if**(flag==1) response.sendRedirect("teacher.jsp");

}**else** {

**try** {

String sql = "SELECT \* FROM dbt\_student";

PreparedStatement pstmt = dbconn.prepareStatement(sql);

ResultSet rst= pstmt.executeQuery();

**while** (rst.next()) {

**if**(num.equals(rst.getString("num"))&&psw.equals(rst.getString("password"))) {

flag=1;

message = "<li>登录成功！</li>";

Student student=**new** Student(rst.getString("name"),rst.getString("ID"),num,psw,rst.getString("institute"),rst.getString("major"),rst.getString("class"));

request.getSession().setAttribute("student",student);

**break**;

}

}

**if**(flag==0) {

message = "<li>登录失败！</li>";

}

}**catch** (SQLException e) {

e.printStackTrace();

message = "<li>登录异常！</li>";

}

request.getSession().setAttribute("result", message);

**if**(flag==1) response.sendRedirect("student.jsp");

}

**if**(flag==0) response.sendRedirect("error.jsp");

}

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

**try** {

dbconn.close();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}