# 实验一 基于Servlet的CRUD

一、实验目的：

熟悉并掌握运用Servlet开发模式 (HTML+JSP+JavaBean+Servlet)即MVC模式实现数据库CRUD基本编程。

二、实验内容：

在MyEclipse环境下运用HTML+JSP+JavaBean+Servlet以及JDBC技术完成相应的数据库CRUD功能，调试运行程序。

三、实验要求：

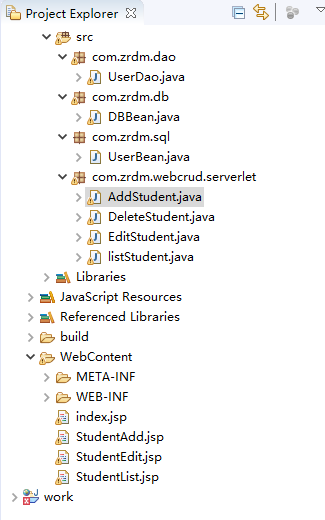
1. 熟悉并掌握运用MVC技术开发功能模块的基本步骤；

2. 运用相关技术（HTML+JSP+JavaBean+Servlet以及JDBC）完成规定功能；

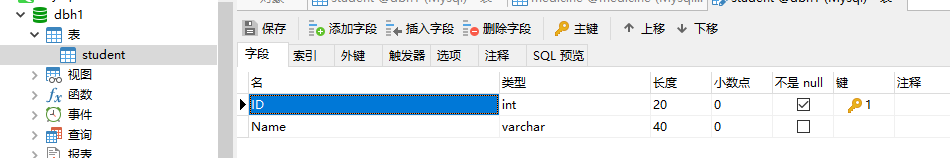
3. 写出实验报告。

四、实验步骤：

1．进入MyEclipse环境，新建一个Web Project。



2. 设计数据库表结构。



3. 采用MVC技术完成数据库的CRUD

// 删除个人信息

**public** **void** delete(**int** id) {

Connection connection = (Connection) DBBean.*getConnection*();

String sql = "delete from student where ID =" + id;

PreparedStatement preparedStatement = **null**;

**try** {

preparedStatement = (PreparedStatement) connection.prepareStatement(sql);

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

preparedStatement.executeUpdate();

} **catch** (SQLException e) {

e.printStackTrace();

} **finally** {

DBBean.*close*(preparedStatement);

DBBean.*close*(connection);

}

}

// 添加个人信息

@SuppressWarnings("deprecation")

**public** **void** add(UserBean userBean) {

String sql = "insert into student(Name,ID) values('" +

userBean.getUsername() + "',"+userBean.getUserid()+");";

System.***out***.println( "\t" + sql);

Connection con = DBBean.*getConnection*();

PreparedStatement preparedStatement = **null**;

//Statement state = null;

**try** {

preparedStatement = (PreparedStatement) con.prepareStatement(sql);

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

preparedStatement.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

DBBean.*close*(preparedStatement);

DBBean.*close*(con);

}

}

// 更新个人信息

**public** **void** update(**int** id,UserBean user) {

String sql = "update student set Name='" + user.getUsername() + "'where ID= "+id;

Connection connection = (Connection) DBBean.*getConnection*();

PreparedStatement preparedStatement = **null**;

**try** {

preparedStatement = (PreparedStatement) connection.prepareStatement(sql);

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

preparedStatement.executeUpdate();

} **catch** (SQLException e) {

e.printStackTrace();

} **finally** {

DBBean.*close*(preparedStatement);

DBBean.*close*(connection);

}

}

/\*\*

\* 列出全部信息

\*

\* **@return**

\*/

**public** List<UserBean> getList() {

String sql = "select \* from student";

Connection connection = (Connection) DBBean.*getConnection*();

PreparedStatement preparedStatement = **null**;

ResultSet resultSet = **null**;

List<UserBean> userBeans = **new** ArrayList<UserBean>();

**try** {

preparedStatement = (PreparedStatement) connection.prepareStatement(sql);

resultSet = preparedStatement.executeQuery();

**while** (resultSet.next()) {

**int** id = resultSet.getInt("ID");

String Name = resultSet.getString("Name");

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

userBeans.add(**new** UserBean(id,Name));

}

} **catch** (SQLException e) {

e.printStackTrace();

} **finally** {

DBBean.*close*(preparedStatement);

DBBean.*close*(connection);

}

**return** userBeans;

}

// 更新个人信息

**public** **void** update(**int** id,UserBean user) {

String sql = "update student set Name='" + user.getUsername() + "'where ID= "+id;

Connection connection = (Connection) DBBean.*getConnection*();

PreparedStatement preparedStatement = **null**;

**try** {

preparedStatement = (PreparedStatement) connection.prepareStatement(sql);

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

preparedStatement.executeUpdate();

} **catch** (SQLException e) {

e.printStackTrace();

} **finally** {

DBBean.*close*(preparedStatement);

DBBean.*close*(connection);

}

}

**public** List<UserBean> getList() {//列出

String sql = "select \* from student";

Connection connection = (Connection) DBBean.*getConnection*();

PreparedStatement preparedStatement = **null**;

ResultSet resultSet = **null**;

List<UserBean> userBeans = **new** ArrayList<UserBean>();

**try** {

preparedStatement = (PreparedStatement) connection.prepareStatement(sql);

resultSet = preparedStatement.executeQuery();

**while** (resultSet.next()) {

**int** id = resultSet.getInt("ID");

String Name = resultSet.getString("Name");

userBeans.add(**new** UserBean(id,Name));

}

} **catch** (SQLException e) {

e.printStackTrace();

} **finally** {

DBBean.*close*(preparedStatement);

DBBean.*close*(connection);

}

**return** userBeans;

}

package com.zrdm.webcrud.serverlet;

import java.io.IOException;

import java.util.Date;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.zrdm.sql.\*;

import com.zrdm.dao.\*;

/\*\*

\* Servlet implementation class AddStudent

\*/

@WebServlet("/AddStudent")

public class AddStudent extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AddStudent() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

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

\*/

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("utf-8");

/\*

\* 每一个调用次 Servlet 的地方都需要传递一个method参数用来指明所要经进行的操作

\*/

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

HttpSession session = request.getSession();

// 根据method的值调用不同的方法执行不同的操作

if ("add".equals(method)) {// 修改药品信息

add(request, response);

}

}

/\*\*

\* 添加药品信息

\*/

private void add(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

System.out.println(new Date().toString() + "\tMedicineServlet.add()开始执行");

req.setCharacterEncoding("utf-8");

UserDao userdao = new UserDao();

String name = req.getParameter("username");

int id =Integer.parseInt( req.getParameter("ID"));

UserBean user=new UserBean();

user.setUsername(name);

user.setUserid(id);

userdao.add(user);

System.out.println("添加成功");

req.getRequestDispatcher("StudentList.jsp").forward(req, resp);

}

}

**private** **void** add(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {

System.***out***.println(**new** Date().toString() + "\tMedicineServlet.add()开始执行");

req.setCharacterEncoding("utf-8");

UserDao userdao = **new** UserDao();

String name = req.getParameter("username");

**int** id =Integer.*parseInt*( req.getParameter("ID"));

UserBean user=**new** UserBean();

user.setUsername(name);

user.setUserid(id);

userdao.add(user);

System.***out***.println("添加成功");

req.getRequestDispatcher("StudentList.jsp").forward(req, resp);

}

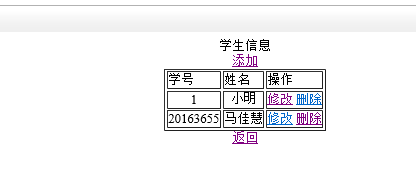
}

4. 程序运行截图。

运行结果：



**图1：首页图**

****

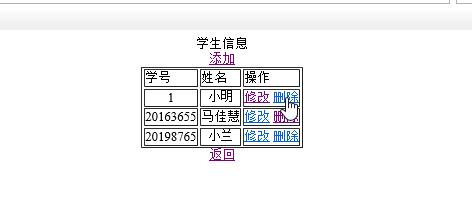
**图2：学生全部信息**

****

**图3：添加学生**

****

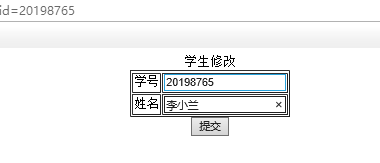
**图4：添加学生展示结果**

****

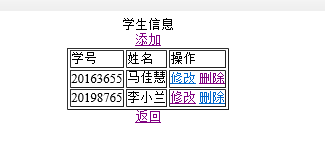
**图5：进行删除**

****

**图6：删除结果**

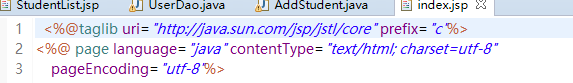
****

**图7：修改信息**

****

**图8：修改信息结果展示**

五、心得体会：

之前对Servlet有一些了解，也做过一些项目，可能时间过得有点久了，这次的实验做起来出下了很多错误，一开始没有导包，也没写使用list的语句：

导致查看所有信息时出不来，这个错误改了小半天，一直怀疑是不是自己哪的代码写错了，一直测试代码，就是改不出错误来，后来发现是自己没有引入上面那句话，后来就是没有把jst的包放Lib下面，而是直接buildpath的，以至于又报错：

The absolute uri: [http://java.sun.com/jsp/jstl/core] cannot be resolved in either web.xml or the jar files deployed with this application。

总之这次的实验帮助我回顾了很多，很感谢这次的实验吧。把servlet掌握的更牢固了，这错误没有下次了。加油！