# 实验二 基于Struts2的CRUD

一、实验目的：

熟悉Struts2 Action类的基本用法，掌握Struts2 Action类在MVC程序开发中的编写和应用，实现CRUD编程。

二、实验内容：

在MyEclipse环境下运用Struts2技术，完成数据库CRUD功能，调试运行程序。

三、实验要求：

1. 掌握运用Struts2 开发MVC程序的基本步骤；

2. 掌握Struts2 Action接收请求参数、调用业务逻辑、封装结果数据并转向结果页面的用法

3. 掌握在struts.xml文件中对Struts2 Action的基本配置；

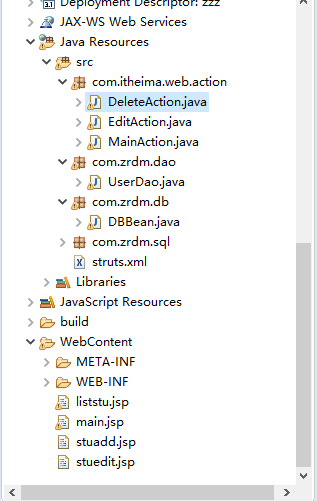
4. 运用相关技术（Struts2以及JDBC等）完成规定功能；

5. 写出实验报告。

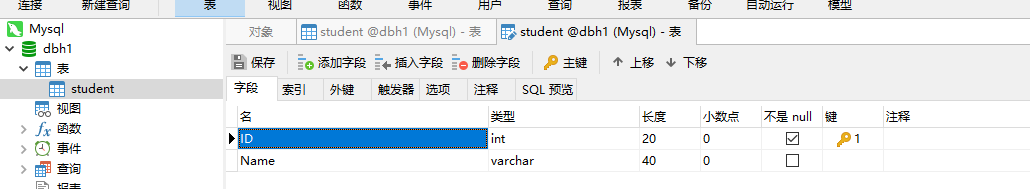
四、实验步骤：

1．进入MyEclipse环境，新建一个Web Project。

项目结构：



2. 设计数据库表结构。



3. 采用Struts2技术完成数据库的CRUD。

核心源码：

**public** **class** MainAction **extends** ActionSupport **implements** ServletRequestAware {

**private** HttpServletRequest request;

**private** UserDao us = **new** UserDao();

**private** UserBean user;

**private** List<UserBean> list;

**private** String id;

**private** String username;

**public** String getId() {

**return** id;

}

**public** **void** setId(String id) {

**this**.id = id;

}

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String add(){

user=**new** UserBean();

user.setUsername(username);

user.setUserid(Integer.*parseInt*(id));

us.add(user);

**return** "success";

}

**public** String list(){

user=**new** UserBean();

list = us.getList();

List<UserBean> students = list;

HttpServletRequest request = ServletActionContext.*getRequest*();

request.setAttribute("students", students);

**return** "success";

}

}

package com.itheima.web.action;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts2.ServletActionContext;

import com.zrdm.dao.UserDao;

import com.zrdm.sql.UserBean;

/\*\*

\* 我们的第一个动作类。

\* 动作类：

\* 它就是一个概念。它就是struts2框架中用于处理请求的类。

\* 我们以后处理请求都写动作类。

\* @author zhy

\*

\*/

public class DeleteAction {

private HttpServletRequest request;

private UserDao us = new UserDao();

private int stId;

public int getStId() {

return stId;

}

public void setStId(int stId) {

this.stId = stId;

}

@SuppressWarnings("static-access")

public String delete() throws Exception {

us.delete(stId);

System.out.println("删除方法执行了。。。。"+this);

return "success";

}

}

<center>

学生信息<br>

<a href="stuadd.jsp">添加学生</a>

<table border=1 >

<tr>

<td>学号</td>

<td>姓名</td>

<td>操作</td>

</tr>

<s:iterator value="#request.students" var="student">

<tr align="center">

<td>${student.userid}</td>

<td>${student.username}</td>

<td align="center"><a href="stuedit.jsp?id=${student.userid}">修改</a>

<a href="${pageContext.request.contextPath}/delete.action?stId=${student.userid}">删除</a></td>

</tr>

</s:iterator>

</table>

<a href="main.jsp">返回</a>

</center>

<struts>

<package name="p1" extends="struts-default">

<action name="stuadd" class="com.itheima.web.action.MainAction" method="add">

<result name="success" type="dispatcher">main.jsp</result>

</action>

<action name="stuedit" class="com.itheima.web.action.EditAction" method="edit">

<result name="success" type="dispatcher">main.jsp</result>

</action>

<action name="list" class="com.itheima.web.action.MainAction" method="list">

<result name="success">liststu.jsp</result>

</action>

<action name="delete" class="com.itheima.web.action.DeleteAction" method="delete">

<result name="success">main.jsp</result>

</action>

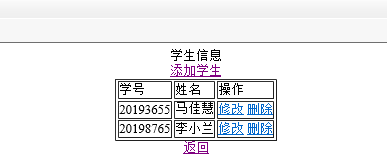
</package>

</struts>

4. 程序运行截图。



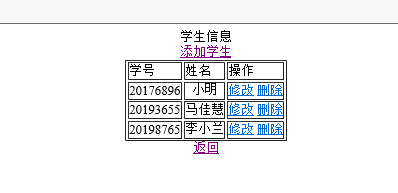
**图1：首页图**

****

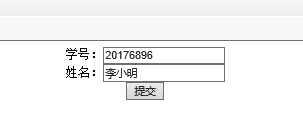
**图2：查看所有学生信息**

****

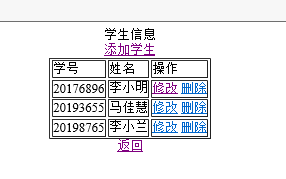
**图3：添加学生**

****

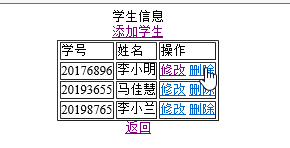
**图4：添加学生结果**

****

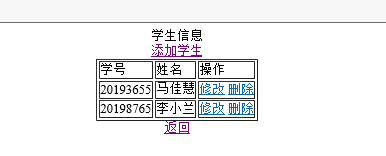
**图5：修改信息**

****

**图6：修改信息结果**

****

**图7：删除学生**

****

**图8：删除学生信息结果**

五、心得体会：

这次的实验我觉得不是很难，但是也不是不难，难就难在我不会操作struts2,所以对整体的代码的结构，传参之间一点也不了解。参考了网上很多例子，似懂非懂的样子就开始了这次实验，在摸索中改bug。很庆幸，最终还是很顺利完成了。下面是我自己总结的，也可能不对，还请老师查看后予以批评指正。

