# 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

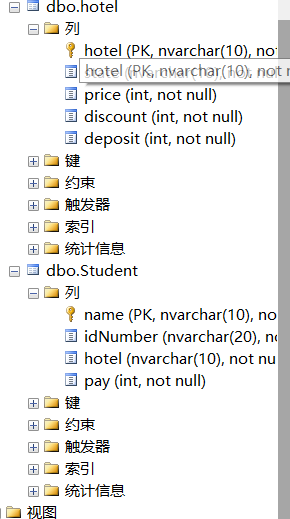
3. 掌握在struts.xml文件中对Struts2 Action的基本配置；

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。

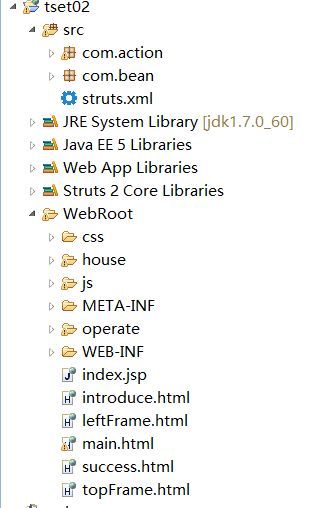
1. 实验步骤：

1.设计数据库表结构。

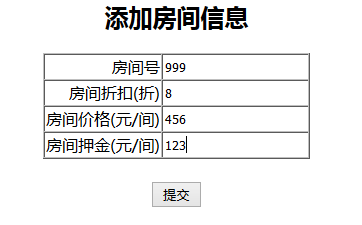


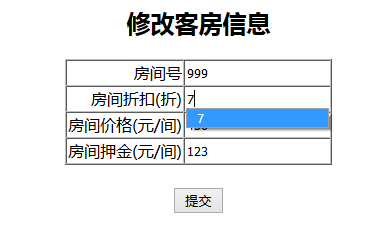
2.采用Struts2技术完成数据库的CRUD。

**总目录结构：**



2.程序运行截图。









1. 关键代码

**Struts.xml**

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

<!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 2.1//EN" "http://struts.apache.org/dtds/struts-2.1.dtd">

<struts>

<package name=*"house"* namespace=*"/house"* extends=*"struts-default"*>

<action name=*"addHouse"* class=*"com.action.AddHouse"*>

<result name=*"success"*>../success.html</result>

<result name=*"error"*>../error.html</result>

</action>

<action name=*"findHouse"* class=*"com.action.FindHouse"*>

<result name=*"success"*>findHouse.jsp</result>

<result name=*"error"*>../error.html</result>

</action>

<action name=*"updateHouse"* class=*"com.action.UpdateHouse"*>

<result name=*"success"*>../success.html</result>

<result name=*"error"*>../error.html</result>

</action>

<action name=*"deleteHouse"* class=*"com.action.DeleteHouse"*>

<result name=*"success"*>../success.html</result>

<result name=*"error"*>../error.html</result>

</action>

</package>

</struts>

**com.action.AddHouse.java**

**package** com.action;

**import** com.bean.DBBean;

**import** com.opensymphony.xwork2.ActionSupport;

**public** **class** AddHouse **extends** ActionSupport {

**private** String id;

**private** String discount;

**private** String price;

**private** String deposit;

**public** String getId() {

**return** id;

}

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

**this**.id = id;

}

**public** String getDiscount() {

**return** discount;

}

**public** **void** setDiscount(String discount) {

**this**.discount = discount;

}

**public** String getPrice() {

**return** price;

}

**public** **void** setPrice(String price) {

**this**.price = price;

}

**public** String getDeposit() {

**return** deposit;

}

**public** **void** setDeposit(String deposit) {

**this**.deposit = deposit;

}

**public** String execute() **throws** Exception

{

**try**

{

String sql="insert into hotel values('"+**this**.getId()+"','未 入住','"+Integer.*parseInt*(**this**.getPrice())+"'" +

",'"+Integer.*parseInt*(**this**.getDiscount())+"','"+Integer.*parseInt*(**this**.getDeposit())+"')";

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

DBBean db=**new** DBBean();

db.executeUpdate(sql);

db.close();

**return** "success";

}

**catch** (Exception e) {

// **TODO**: handle exception

**return** "error";

}

}

}

**com.bean.DBBean.java**

**package** com.bean;

**import** java.sql.\*;

**public** **class** DBBean {

**private** String driverStr = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

**private** String connStr = "jdbc:sqlserver://localhost:1433; DatabaseName=hotelInfo";

**private** String dbusername = "guoguo";

**private** String dbpassword = "123456";

**private** Connection conn = **null**;

**private** Statement stmt = **null**;

**public** DBBean() {

**try** {

Class.*forName*(driverStr);

conn = DriverManager.*getConnection*(connStr, dbusername, dbpassword);

stmt = conn.createStatement();

} **catch** (Exception ex) {

System.*out*.println(ex.getMessage());

}

}

**public** **int** executeUpdate(String s) {

**int** result = 0;

**try** {

result = stmt.executeUpdate(s);

} **catch** (Exception ex) {

System.*out*.println("执行更新错误:"+ex);

}

**return** result;

}

**public** ResultSet executeQuery(String s) {

ResultSet rs = **null**;

**try** {

rs = stmt.executeQuery(s);

} **catch** (Exception ex) {

System.*out*.println("执行查询错误！");

}

**return** rs;

}

**public** **void** close() {

**try** {

stmt.close();

conn.close();

} **catch** (Exception e) {

}

}

}

五、心得体会：

Struts2代替了servlet实现了控制器的作用，而且struts提供的值栈获取jsp的值更加的简单，方便。