# 实验三 基于Hibernate的CRUD

一、实验目的：

熟悉并掌握运用Servlet开发模式(HTML+JSP+JavaBean+Servlet)和Hibernate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Hibernate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技术和Hibernate技术开发功能模块的基本步骤；

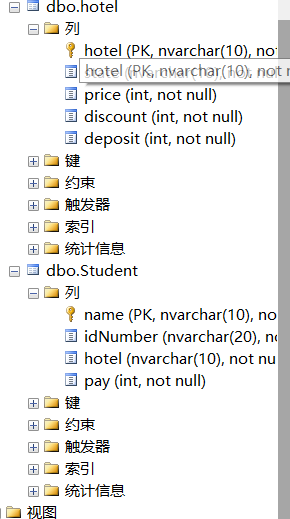
2. 运用相关技术（HTML+JSP+JavaBean+Servlet以及Hibernate）完成规定功能；

3. 写出实验报告。

四、实验步骤：

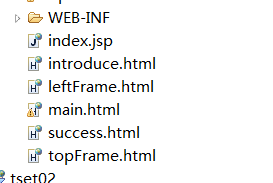
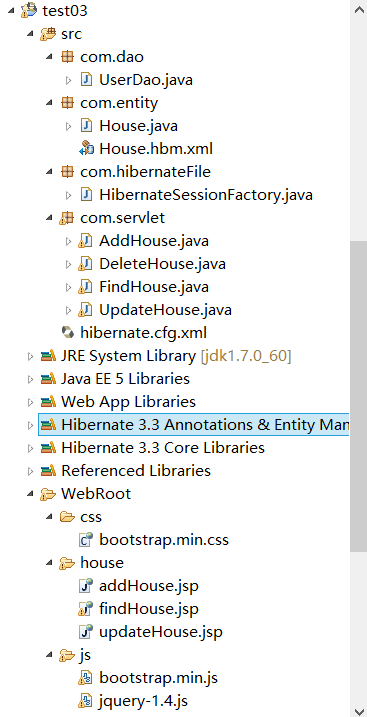
1.设计数据库表结构。

结构如下：

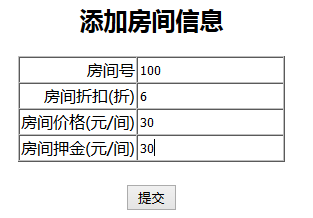


2. 采用MVC技术和Hibernate完成数据库的CRUD。

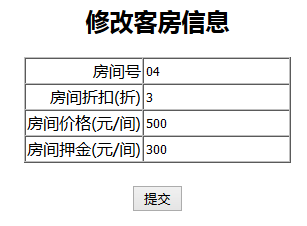
总目录结构：



3.程序运行截图。











4.关键代码：

com.dao.UserDao

package com.dao;

import com.entity.House;

import com.hibernateFile.HibernateSessionFactory;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.Transaction;

public class UserDao {

private Transaction transaction;

private Session session;

private Query query;

public boolean saveInfo(Object obj)

{

try

{

session=HibernateSessionFactory.getSession();

transaction=session.beginTransaction();

session.saveOrUpdate(obj);

transaction.commit();

session.close();

return true;

}

catch(Exception e)

{

e.printStackTrace();

return false;

}

}

public boolean deleteInfo(Object obj)

{

try

{

session=HibernateSessionFactory.getSession();

transaction=session.beginTransaction();

session.delete(obj);

transaction.commit();

session.close();

return true;

}

catch(Exception e)

{

e.printStackTrace();

return false;

}

}

@SuppressWarnings("unchecked")

public List<House> findAll()

{

session=HibernateSessionFactory.getSession();

try

{

transaction=session.beginTransaction();

//HQL语句

String queryString="";

queryString="from House as model";

query=session.createQuery(queryString);

List<House> list=(List<House>) query.list();

transaction.commit();

session.close();

return list;

}

catch(Exception e)

{

e.printStackTrace();

return null;

}

}

@SuppressWarnings("unchecked")

public List<House> findById( String value)

{

session=HibernateSessionFactory.getSession();

try

{

transaction=session.beginTransaction();

//HQL语句

String queryString="";

queryString="from House as model where model.id='"+value+"'";

query=session.createQuery(queryString);

List<House> list=(List<House>) query.list();

transaction.commit();

session.close();

return list;

}

catch(Exception e)

{

e.printStackTrace();

return null;

}

}

}

com.servlet.AddHouse

package com.servlet;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.dao.UserDao;

import com.entity.House;

public class AddHouse extends HttpServlet {

protected void processRequest(HttpServletRequest request,HttpServletResponse response)

throws ServletException,IOException

{

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

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

int price=Integer.parseInt(request.getParameter("price"));

int discount=Integer.parseInt(request.getParameter("discount"));

int deposit=Integer.parseInt(request.getParameter("deposit"));

House house=new House();

house.setHouseId(id);

house.setHousePrice(price);

house.setHouseDiscount(discount);

house.setHouseState("未入住");

house.setHouseDeposit(deposit);

PrintWriter out=response.getWriter();

UserDao dao=new UserDao();

try

{

dao.saveInfo(house);

out.println("<script language='javaScript'> alert('添加成功!');</script>");

response.setHeader("refresh", "1;url=findHouseServlet?type=find");

out.close();

} catch (Exception e)

{

// TODO: handle exception

e.printStackTrace();

out.println("<script language='javaScript'> alert('添加失败!');</script>");

response.setHeader("refresh", "1;url=house/addHouse.jsp");

}

}

protected void doGet(HttpServletRequest request,HttpServletResponse response)

throws ServletException,IOException

{

processRequest(request, response);

}

protected void doPost(HttpServletRequest request,HttpServletResponse response)

throws ServletException,IOException

{

processRequest(request, response);

}

}

com.hibernameFile.HibernateSessionFactory

package com.hibernateFile;

import javax.swing.JOptionPane;

import org.hibernate.HibernateException;

import org.hibernate.Session;

import org.hibernate.cfg.Configuration;

import org.hibernate.SessionFactory;

public class HibernateSessionFactory {

private static SessionFactory sessionFactory;

private static Configuration configuration = new Configuration();

private HibernateSessionFactory() {

}

static {

try

{

Configuration configure=configuration.configure("hibernate.cfg.xml");

sessionFactory=configure.buildSessionFactory();

}

catch(Exception e)

{

message("生成SessionFactory失败:"+e);

}

}

public static Session getSession() throws HibernateException {

return sessionFactory.openSession();

}

private static void message(String mess) {

// TODO Auto-generated method stub

int type=JOptionPane.YES\_NO\_OPTION;

String title="提示信息";

JOptionPane.showMessageDialog(null, mess, title, type);

}

}

com.servlet.FindHouse

package com.servlet;

import java.io.IOException;

import java.util.List;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.dao.UserDao;

import com.entity.House;

public class FindHouse extends HttpServlet {

protected void processRequest(HttpServletRequest request,HttpServletResponse response)

throws ServletException,IOException

{

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

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

System.out.println(type);

UserDao dao=new UserDao();

try

{

List<House> list=dao.findAll();

System.out.println(list);

request.setAttribute("list", list);

request.getRequestDispatcher("house/findHouse.jsp").forward(request, response);

/\*if(type.equals("checkIn"))

{

request.getRequestDispatcher("operate/checkIn.jsp").forward(request, response);

}

else if(type.equals("checkOut"))

{

request.getRequestDispatcher("operate/checkOut.jsp").forward(request, response);

}

else

{

request.getRequestDispatcher("house/findHouse.jsp").forward(request, response);

}\*/

}

catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

response.getWriter().println("<script language='javaScript'> alert('查询失败!');</script>");

}

}

protected void doGet(HttpServletRequest request,HttpServletResponse response)

throws ServletException,IOException

{

processRequest(request, response);

}

protected void doPost(HttpServletRequest request,HttpServletResponse response)

throws ServletException,IOException

{

processRequest(request, response);

}

}

1. 心得体会：

Hibernate是一个开放源代码的对象关系映射框架，它对JDBC进行了非常轻量级的对象封装，它将POJO与数据库表建立映射关系，是一个全自动的orm框架，hibernate可以自动生成SQL语句，自动执行，使得Java程序员可以随心所欲的使用对象编程思维来操纵数据库。 Hibernate可以应用在任何使用JDBC的场合，既可以在Java的客户端程序使用，也可以在Servlet/JSP的Web应用中使用，最具革命意义的是，Hibernate可以在应用EJB的J2EE架构中取代CMP，完成数据持久化的重任。