# 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；

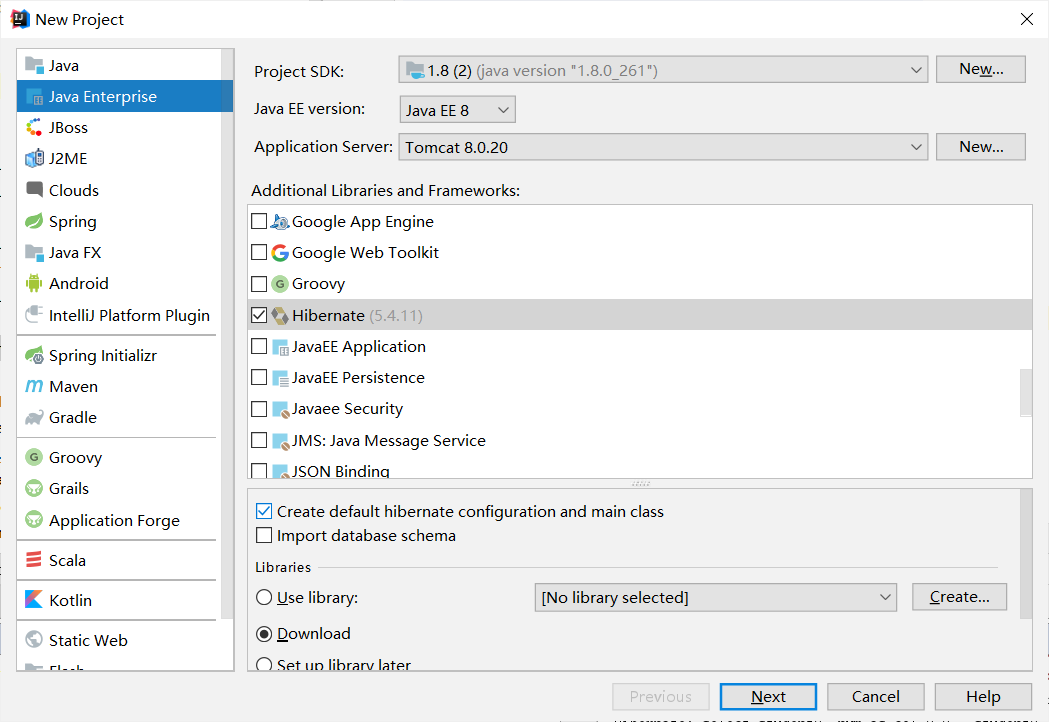
3. 写出实验报告。

四、实验步骤：

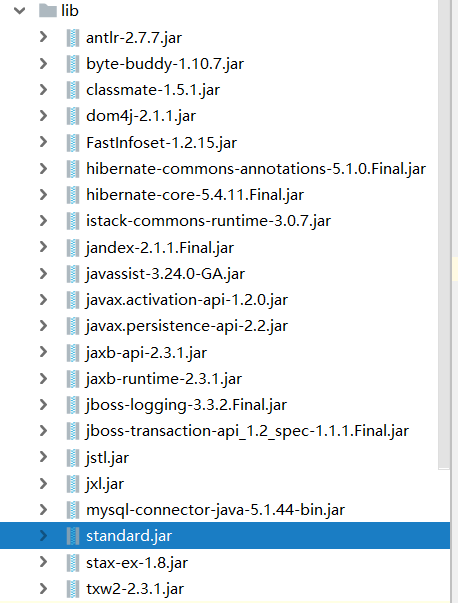
1．进入MyEclipse环境，新建一个Web Project。

我这里用的是IDEA环境，新建Java Enterprise项目

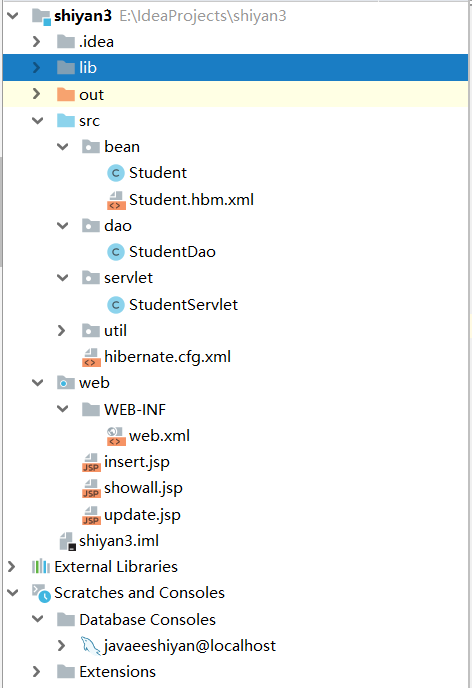
勾选Web Application和Hibernate



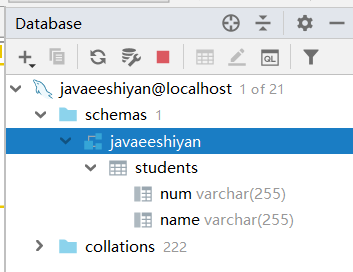
导入所需的jar包



项目目录



1. 设计数据库表结构。



1. 采用MVC技术和Hibernate完成数据库的CRUD。

详细代码：

Student.java

**package** bean;  
  
**import** java.io.Serializable;  
  
**public class** Student **implements** Serializable {  
 **private** String **num**;  
 **private** String **name**;  
  
 **public** Student() {  
 **super**();  
 *//* ***TODO Auto-generated constructor stub*** }  
  
 **public** Student(String name, String num) {  
 **super**();  
 **this**.**num** = num;  
 **this**.**name** = name;  
 }  
 **public** String getNum() {  
 **return num**;  
 }  
  
 **public void** setNum(String num) {  
 **this**.**num** = num;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
}

Student.hbm.xml

*<?***xml version="1.0" encoding="UTF-8"** *?>***<!DOCTYPE hibernate-mapping PUBLIC  
 "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  
 "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd"*>***<**hibernate-mapping**>  
 <**class name="bean.Student" table="students"**>  
 <**composite-id**>  
 <**key-property name="num" type="java.lang.String"**>  
 <**column name="num"** />  
 </**key-property**>  
 <**key-property name="name" type="java.lang.String"**>  
 <**column name="name"** />  
 </**key-property**>  
  
 </**composite-id**>  
 </**class**>  
</**hibernate-mapping**>

StudentDao.java

**package** dao;  
  
**import** bean.Student;  
**import** org.hibernate.Session;  
**import** org.hibernate.Transaction;  
**import** util.unitl;  
  
**import** java.util.List;  
  
**public class** StudentDao {  
 *//调用工具包* **static** util.unitl *unitl*=**new** unitl();  
 **static** Session *session*=*unitl*.lj();  
 **int a**=0;  
  
 */\*\*\*\*\*\*\*\*\*\*\*添加学生\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/* **public int** addstudent(String num,String name){  
 **int** a=0;  
 *//3、开启事务* Transaction ts=*session*.beginTransaction();  
 *//执行数据库操作  
 //添加操作* Student student=**new** Student();  
 student.setNum(num);  
 student.setName(name);  
 *session*.save(student);  
 *//提交* ts.commit();  
 *//关闭session  
 //session.close();* a=1;  
 **return** a;  
 }  
  
 */\*\*\*\*\*\*\*\*\*\*\*查看所有学生\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/* @SuppressWarnings(**"unchecked"**)  
 **public** List<Student> selectall(){  
 *//执行数据库操作  
 //编写hql* String hql=**"from Student "**;  
 List<Student> list=*session*.createQuery(hql).list();  
 **return** list;  
 }  
 */\*\*\*\*\*\*\*\*\*\*\*根据学号删除学生\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/* **public int** delete(String num){  
 **int** a=0;  
 *//3、开启事务* Transaction ts=*session*.beginTransaction();  
 *//执行数据库操作  
 //删除操作  
 session*.createQuery(**"delete from Student where num = ?0"**)  
 .setString(0, num).executeUpdate() ;  
 *//提交* ts.commit();  
 *//关闭session  
 //session.close();* a=1;  
 **return** a;  
 }  
 */\*\*\*\*\*\*\*\*\*\*\*根据ID查询一个部门\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/* **public** List<Student> selectone(String num){  
 *//执行数据库操作  
 //编写hql* String hql=**"select new Student (num,name) from Student where num=?0"**;  
 List<Student> list=*session*.createQuery(hql)  
 .setString(0, num)  
 .list();  
 **return** list;  
 }  
 */\*\*\*\*\*\*\*\*\*\*\*修改部门\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/* **public void** update(String num, String name){  
 *//执行数据库操作  
 //编写hql* Transaction ts=*session*.beginTransaction();  
 *session*.createQuery(**"update Student s set s.name = ?1 where num =?0"**)  
 .setString(0, num).setString(1,name).executeUpdate() ;  
 *//提交* ts.commit();  
 }  
  
}

StudentServlet.java

**package** servlet;  
  
**import** bean.Student;  
**import** dao.StudentDao;  
  
**import** javax.servlet.RequestDispatcher;  
**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** java.io.IOException;  
**import** java.io.PrintWriter;  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
@WebServlet(name = **"StudentServlet"**,urlPatterns = **"/StudentServlet"**)  
**public class** StudentServlet **extends** HttpServlet {  
 StudentDao **sd**=**new** StudentDao();  
 **public** StudentServlet(){  
 **super**();  
 }  
 **public void** destroy() {  
 **super**.destroy(); *// Just puts "destroy" string in log  
 // Put your code here* }  
  
 **protected void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  
 request.setCharacterEncoding(**"utf-8"**);  
 response.setCharacterEncoding(**"utf-8"**);  
 PrintWriter out = response.getWriter();  
 *//用type接受页面的值* String type =**""**;  
 type = request.getParameter(**"type"**);  
 **if** (**"1"**.equals(type)) { *//添加学生的方法* addStudent(request, response);  
 }**else if** (**""**.equals(type)||**null**==type) { *//查询所有部门的方法* selectall(request, response);  
 }**else if** (**"3"**.equals(type)) { *//根据ID删除部门* deletestudent(request, response);  
 }**else if** (**"4"**.equals(type)) { *//根据修改部门* update(request, response);  
 }**else if** (**"5"**.equals(type)) { *//根据ID查询一个部门* selectone(request, response);  
 }  
 out.flush();  
 out.close();  
 }  
  
 **protected void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  
 request.setCharacterEncoding(**"utf-8"**);  
 response.setCharacterEncoding(**"utf-8"**);  
 response.setContentType(**"text/html;charset=utf-8"**);  
 PrintWriter out = response.getWriter();  
 **this**.doPost(request, response);  
 out.flush();  
 out.close();  
 }  
 */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*添加学生\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/* **public void** addStudent(HttpServletRequest request, HttpServletResponse response)  
 **throws** ServletException, IOException {  
 String num=request.getParameter(**"num"**);  
 String name=request.getParameter(**"name"**);  
 **int** a=**sd**.addstudent(num,name);  
 **if** (a>0) {  
 selectall(request, response);  
 System.***out***.println(**"添加部门成功"**);  
 }**else** {  
 System.***out***.println(**"添加部门失败"**);  
 response.sendRedirect(**"insert.jsp"**);  
 }  
 }  
 */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*查询所有学生\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/* **public void** selectall(HttpServletRequest request, HttpServletResponse response)  
 **throws** ServletException, IOException {  
 List<Student> list=**new** ArrayList<Student>();  
 list=**sd**.selectall();  
 request.setAttribute(**"list"**, list);  
 RequestDispatcher rd=request.getRequestDispatcher(**"showall.jsp"**);  
 rd.forward(request, response);  
 }  
 */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*根据ID删除部门\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/* **public void** deletestudent(HttpServletRequest request, HttpServletResponse response)  
 **throws** ServletException, IOException {  
 String num=request.getParameter(**"num"**);  
 **sd**.delete(num);  
 selectall(request, response);  
 }  
 */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*根据学号查找学生\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/* **public void** selectone(HttpServletRequest request, HttpServletResponse response)  
 **throws** ServletException, IOException {  
 String num=request.getParameter(**"num"**);  
 List<Student> list=**sd**.selectone(num);  
 **for**(Student stu:list){  
 *//System.out.println(bb.getBmname()+bb.getId());* request.setAttribute(**"list"**, stu);  
 }  
 *//System.out.println("aa");* RequestDispatcher rd=request.getRequestDispatcher(**"update.jsp"**);  
 rd.forward(request, response);  
 }  
 */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*修改学生\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/* **public void** update(HttpServletRequest request, HttpServletResponse response)  
 **throws** ServletException, IOException {  
 String num=request.getParameter(**"num"**);  
 String name=request.getParameter(**"name"**);  
 **sd**.update(num, name);  
 selectall(request, response);  
 }  
 **public void** init() **throws** ServletException {  
 *// Put your code here* }  
  
  
}

Unitl.java

**package** util;  
  
**import** org.hibernate.Session;  
**import** org.hibernate.SessionFactory;  
**import** org.hibernate.boot.registry.StandardServiceRegistry;  
**import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;  
**import** org.hibernate.cfg.Configuration;  
**import** org.hibernate.service.ServiceRegistry;  
  
**public class** unitl {  
  
 **public** Session lj(){  
 Configuration configure = **new** Configuration().configure();  
 StandardServiceRegistry serviceRegistry = **new** StandardServiceRegistryBuilder().configure().build();  
 SessionFactory sessionFactory = configure.buildSessionFactory(serviceRegistry);  
 Session session = sessionFactory.openSession();  
 **return** session;  
  
 }  
}

hibernate.cfg.xml

*<?***xml version='1.0' encoding='utf-8'***?>***<!DOCTYPE hibernate-configuration PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD//EN"  
 "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd"*>***<**hibernate-configuration**>  
 <**session-factory**>  
 *<!--数据库驱动-->* <**property name="connection.driver\_class"**>com.mysql.jdbc.Driver</**property**>  
 *<!--连接url-->  
 <!--连接url-->* <**property name="connection.url"**>jdbc:mysql://localhost:3306/javaeeshiyan?useUnicode=true**&amp;**characterEncoding=UTF-8</**property**>  
 *<!--用户名-->* <**property name="connection.username"**>root</**property**>  
 *<!--密码-->* <**property name="connection.password"**>123456</**property**>  
 *<!--数据库方言-->* <**property name="dialect"**>org.hibernate.dialect.MySQLDialect</**property**>  
 *<!--显示SQL语句-->* <**property name="show\_sql"**>true</**property**>  
 *<!--关系映射-->* <**mapping class="bean.Student"**/>  
 <**mapping resource="bean/Student.hbm.xml"**/>  
 </**session-factory**>  
</**hibernate-configuration**>

web.xml

*<?***xml version="1.0" encoding="UTF-8"***?>*<**web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"  
 version="4.0"**>  
 <**welcome-file-list**>  
 <**welcome-file**>StudentServlet</**welcome-file**>  
 </**welcome-file-list**>  
</**web-app**>

showall.jsp

<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**"%>  
<%@**taglib uri**="**http://java.sun.com/jsp/jstl/core**" **prefix**="**c**" %>  
<**html**>  
<**head**>  
 <**title**>学生信息</**title**>  
</**head**>  
<**body**>  
<**center**>  
<**h1**>学生信息</**h1**>  
<**a href="insert.jsp"**>添加</**a**>  
<**table border="1"**>  
 <**tr**>  
 <**td**>学号</**td**>  
 <**td**>姓名</**td**>  
 <**td**>操作</**td**>  
 </**tr**>  
 <**c:forEach items="${**list**}" var="stu"**>  
 <**tr**>  
 <**td**>**${**stu.num**}**</**td**><**td**>**${**stu.name**}**</**td**>  
 <**td**><**a href="StudentServlet?type=5&&num=${**stu.num**}"**>修改</**a**>*<!-- 修改前先查一个 -->*     
 <**a href="StudentServlet?type=3&&num=${**stu.num**}"**>删除</**a**></**td**>  
  
 </**tr**>  
 </**c:forEach**>  
</**table**>  
</**center**>  
</**body**>  
</**html**>

insert.jsp

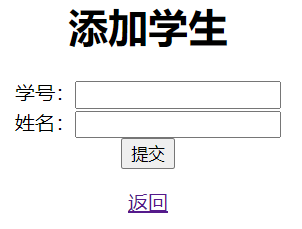
*<%--  
 Created by IntelliJ IDEA.  
 User: ASUS  
 Date: 2021/4/28  
 Time: 19:57  
 To change this template use File | Settings | File Templates.  
--%>*<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
<**head**>  
 <**title**>添加学生</**title**>  
</**head**>  
<**body**>  
<**center**>  
<**h1**>添加学生</**h1**>  
<**form action="StudentServlet" method="post"**>  
 <**input type="hidden" value="1" name="type"**>  
 学号：<**input type="text" name="num"**/><**br**/>  
 姓名：<**input type="text" name="name"**/><**br**/>  
 <**input type="submit" value="提交"**>  
</**form**>  
<**a href="StudentServlet"**>返回</**a**>  
</**center**>  
</**body**>  
  
</**html**>

update.jsp

<%@ **page contentType**="**text/html;charset=UTF-8**" **language**="**java**" %>  
<**html**>  
<**head**>  
 <**title**>修改学生</**title**>  
</**head**>  
<**body**>  
<**center**>  
<**h1**>修改学生</**h1**>  
<**form action="StudentServlet" method="post"**>  
 <**input type="hidden" value="4" name="type"**>  
 学号:<**input type="text" value="${**list.name **}" name="num" readonly**><**br**/>  
 姓名:<**input type="text" name="name" value="${**list.num**}"**><**br**/>  
 <**input type="submit" value="修改"**>  
</**form**>  
 <**a href="StudentServlet"**>返回</**a**>  
</**center**>  
</**body**>  
</**html**>

1. 程序运行截图。











1. 心得体会：

学习了Hibernate框架，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，学习使用hql语句，学习到了很多新的知识。