WEB开发技术

实验指导书

学 院： 计算机科学与工程学院

专 业： 物联网 班 级： 2016-2班

姓 名： 刘东临 学 号： 201601061413

山东科技大学计算机科学与工程学院

二零一八 年 十一 月

# 实验 1 Web前端练习

## 一、实验目的

1． 掌握HTML和CSS的用法。

2． 了解 HTML语言的概貌。

## 二、实验内容

1．请用HTML编写如下的个人简历表格:

|  |  |  |  |
| --- | --- | --- | --- |
| **(基本信息)** | | | *简历更新日期：2009-04-30* |
| **姓名：** | 张三 | |  |
| **身份证号码：** | 32010419781202081x | |
| **性别：** | 男 | |
| **出生年月：** | 1978.12.02 | |
| **民族：** | 汉族 | **政治面貌：** | 共青团员 |
| **婚姻状况：** | 未婚 | **视力：** |  |
| **户籍所在地：** | 南京市秦淮区 | **身高：** | 0(cm) |
| **现居住地：** | 南京市秦淮区 | | |
| **(目前从事工作情况)** | | | |
| **现工作单位：** | \*\*\* | **参加工作年度：** | 1998年 |
| **目前从事岗位：** | 信息工程 | **从事现岗位时间：** | 8年 |
| **现职称级别：** |  | | |
| **(受教育情况)** | | | |
| **学历：** | 本科 | **毕业年度：** | 1998 |
| **所学专业：** | 涉外会计 | | |
| **毕业院校：** | 金陵科技学院 | | |
| **外语及水平：** | 英语，普通英语六级；法语，一般 | | |
| **学习简历：** | 1995/1--1998/9金陵科技学院金融学 | | |
| **继续教育情况：** |  | | |
| **工作简历：** | 2000/3--2008：南京嘉腾科技有限公司 所属行业：计算机软件 EximbillsEnterprise项目经理 1.负责产品国际市场发展环境的分析,包括区域电信发展环境，重点运营商经营发展战略、投资方向和客户需求的分析；侧重海外高端运营商的市场培育、机会挖掘； 2.负责海外片区市场投资空间的中长期预测以及进行所辖区域市场竞争格局的分析.参与市场规划，定期输出所辖区域的市场策略（市场目标选择，市场竞争策略，产品策略以及合作策略等）；为产品的市场规划和重大市场营销活动提供支持 3.在IBMWebsphere平台上用Java设计并开发国际结算系统EximBills企业级版本， 参与系统设计过程，并撰写英文系统文档。用MSProject编写项目计划，沟通各项目经理，海外市场销售团队与公司管理层，控制项目目标与公司整体战略之间的平衡。 4.协调渣塔银行StandardCharteredBankMegellan项目，协助项目经理沟通银行客户，保持中东公司实施团队与南京研发中心系统开发版本的一致。为银行提供系统技术咨询服务并分五个阶段将整个项目 | | |
| **个人技能及特长：** | 1.金融业务理论 2.Java软件开发 3.精通SQLServer开发 3.英语,法语流利 4.有较强的国际视野和跨文化沟通能力。 | | |
| **成果及获奖情况：** |  | | |
| **(求职意向)** | | | |
| **求职意向：** | 信息技术经理/主管，科研人员，科研管理人员 | | |
| **期望工作地点：** | 南京市 | | |
| **薪酬要求：** | 面议 | **工作方式：** | 全/兼职 |
| **对新工作的期望：** |  | | |
| **(其它资料)** | | | |
| **兴趣爱好：** |  | | |
| **自我评价：** | 多年Java软件开发经验，英语,法语流利,有较强的国际视野和跨文化沟通能力。MCSE,MCDBA,MCSD | | |
| **(联系方式)** | | | |
| **联系电话：** | 025-84589702 | **住宅电话：** | 025-84589702 |
| **移动电话：** | 13951725880 | **联系人：** |  |
| **E-MAIL：** | xxxxxx@hotmail.com | **个人主页：** |  |
| **聊天帐号：** |  | | |
| **通讯地址：** | （） | | |

## 3、 实验步骤

代码如下：

<!doctype html>

<html lang="zh">

<head>

<meta charset="UTF-8">

<title>个人简历</title>

</head>

<body>

<table width="800" border="1" align="center" cellpadding="10" cellspacing="0">

<caption><h3>个人简历</h3></caption>

<tr>

<td colspan="3"><font color="blue"><b>(基本信息)</b></font></td>

<td><font color="red"><em>简历更新日期：2009-04-30</em></font></td>

</tr>

<tr>

<td><b>姓 名：</b></td>

<td colspan="2">张三</td>

<td rowspan="4" align="center" valign="middle">照片</td>

</tr>

<tr>

<td><b>身份证号码：</b></td>

<td colspan="2">32010419781202081x</td>

</tr>

<tr>

<td><b>性 别：</b></td>

<td colspan="2">男</td>

</tr>

<tr>

<td><b>出生年月：</b></td>

<td colspan="2">1978.12.02</td>

</tr>

<tr>

<td><b>民族：</b></td>

<td>汉族</td>

<td><b>政治面貌：</b></td>

<td>共青团员</td>

</tr>

<tr>

<td><b>婚姻状况：</b></td>

<td>未婚</td>

<td><b>视力：</b></td>

<td></td>

</tr>

<tr>

<td><b>户籍所在地：</b></td>

<td>南京市秦淮区</td>

<td><b>身高：</b></td>

<td>180（cm）</td>

</tr>

<tr>

<td><b>现居住地：</b></td>

<td colspan="3">南京市秦淮区</td>

</tr>

<tr>

<td colspan="4"><font color="blue"><b>(目前从事工作情况)</b></font></td>

</tr>

<tr>

<td><b>现工作单位：</b></td>

<td>\*\*\*\*\*\*</td>

<td><b>参加工作年度：</b></td>

<td>1998年</td>

</tr>

<tr>

<td><b>目前从事岗位：</b></td>

<td>信息工程</td>

<td><b>从事岗位时间：</b></td>

<td>8年</td>

</tr>

<tr>

<td><b>现职称级别：</b></td>

<td colspan=3></td>

</tr>

<tr>

<td colspan="4"><font color="blue"><b>(受教育情况)</b></font></td>

</tr>

<tr>

<td><b>学历：</b></td>

<td>本科</td>

<td><b>毕业年度：</b></td>

<td>1998年</td>

</tr>

<tr>

<td><b>所学专业：</b></td>

<td colspan=3>涉外会计</td>

</tr>

<tr>

<td><b>毕业院校：</b></td>

<td colspan=3>金陵科技学院</td>

</tr>

<tr>

<td><b>英语及水平：</b></td>

<td colspan=3>英语，普通英语六级；法语，一般</td>

</tr>

<tr>

<td><b>学习简历：</b></td>

<td colspan=3>1995/1--1998/9金陵科技学院金融学</td>

</tr>

<tr>

<td><b>继续教育经历：</b></td>

<td colspan=3></td>

</tr>

<tr>

<td><b>工作简历：</b></td>

<td colspan=3>2000/3--2008：南京嘉腾科技有限公司</br>

所属行业：计算机软件</br>

EximbillsEnterprise项目经理</br>

1.负责产品国际市场发展环境的分析,包括区域电信发展环境，重点运营商经营发展战略、投资方向和客户需求的分析；侧重海外高端运营商的市场培育、机会挖掘；</br>

2.负责海外片区市场投资空间的中长期预测以及进行所辖区域市场竞争格局的分析.参与市场规划，定期输出所辖区域的市场策略（市场目标选择，市场竞争策略，产品策略以及合作策略等）；为产品的市场规划和重大市场营销活动提供支持</br>

3.在IBMWebsphere平台上用Java设计并开发国际结算系统EximBills企业级版本，</br>

参与系统设计过程，并撰写英文系统文档。用MSProject编写项目计划，沟通各项目经理，海外市场销售团队与公司管理层，控制项目目标与公司整体战略之间的平衡。</br>

4.协调渣塔银行StandardCharteredBankMegellan项目，协助项目经理沟通银行客户，保持中东公司实施团队与南京研发中心系统开发版本的一致。为银行提供系统技术咨询服务并分五个阶段将整个项目</br>

</td>

</tr>

<tr>

<td><b>个人技能及特长：</b></td>

<td colspan=3>1.金融业务理论</br>

2.Java软件开发</br>

3.精通SQLServer开发</br>

3.英语,法语流利</br>

4.有较强的国际视野和跨文化沟通能力。</br>

</td>

</tr>

<tr>

<td><b>成果及获奖情况：</b></td>

<td colspan=3></td>

</tr>

<tr>

<td colspan="4"><font color="blue"><b>(求职意向)</b></font></td>

</tr>

<tr>

<td><b>求职意向：</b></td>

<td colspan=3>信息技术经理/主管，科研人员，科研管理人员</td>

</tr>

<tr>

<td><b>期望工作地点：</b></td>

<td colspan=3>南京市</td>

</tr>

<tr>

<td><b>薪酬要求：</b></td>

<td>面议</td>

<td><b>工作方式：</b></td>

<td>全/兼职</td>

</tr>

<tr>

<td><b>对新工作的期望：</b></td>

<td colspan=3></td>

</tr>

<tr>

<td colspan="4"><font color="blue"><b>(其他资料)</b></font></td>

</tr>

<tr>

<td><b>兴趣爱好：</b></td>

<td colspan=3></td>

</tr>

<tr>

<td><b>自我评价：</b></td>

<td colspan=3>多年Java软件开发经验，英语,法语流利,有较强的国际视野和跨文化沟通能力。MCSE,MCDBA,MCSD</td>

</tr>

<tr>

<td colspan="4"><font color="blue"><b>(联系方式)</b></font></td>

</tr>

<tr>

<td><b>联系电话：</b></td>

<td>025-84589702</td>

<td><b>住宅电话：</b></td>

<td>025-84589702</td>

</tr>

<tr>

<td><b>移动电话</b></td>

<td>13951725880</td>

<td><b>联系人：</b></td>

<td></td>

</tr>

<tr>

<td><b>E-MAIL：</b></td>

<td>xxxxxx@hotmail.com</td>

<td><b>个人主页：</b></td>

<td></td>

</tr>

<tr>

<td><b>聊天账号：</b></td>

<td colspan=3></td>

</tr>

<tr>

<td><b>通讯地址：</b></td>

<td colspan=3>（）</td>

</tr>

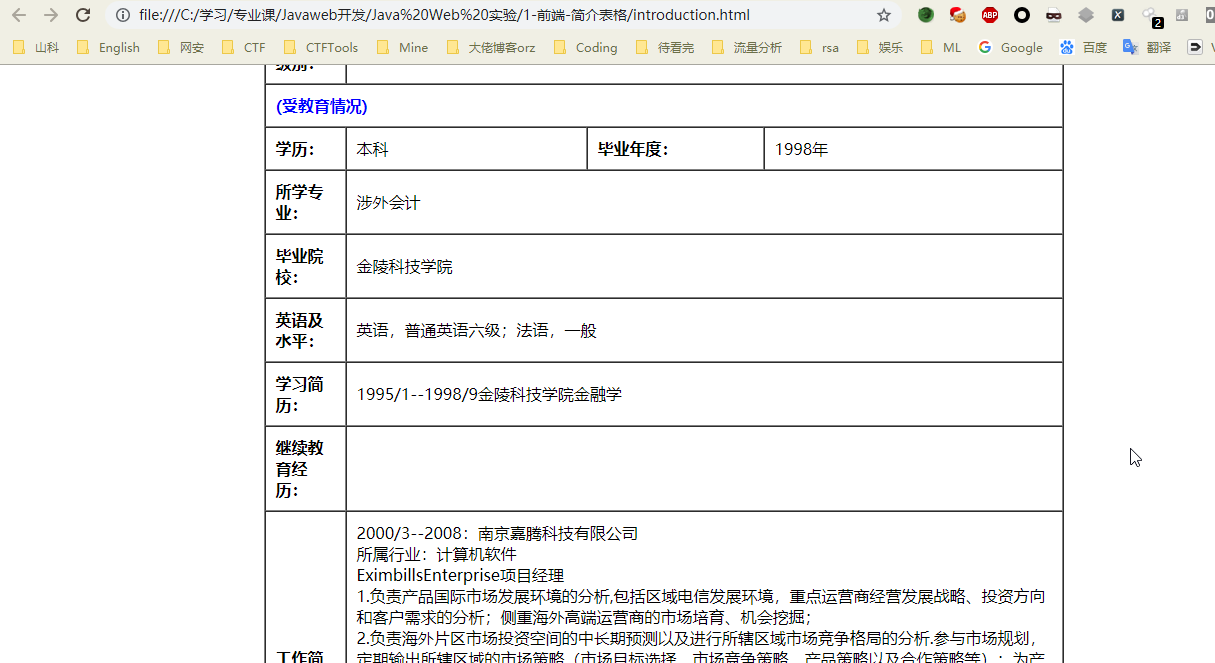
</table>

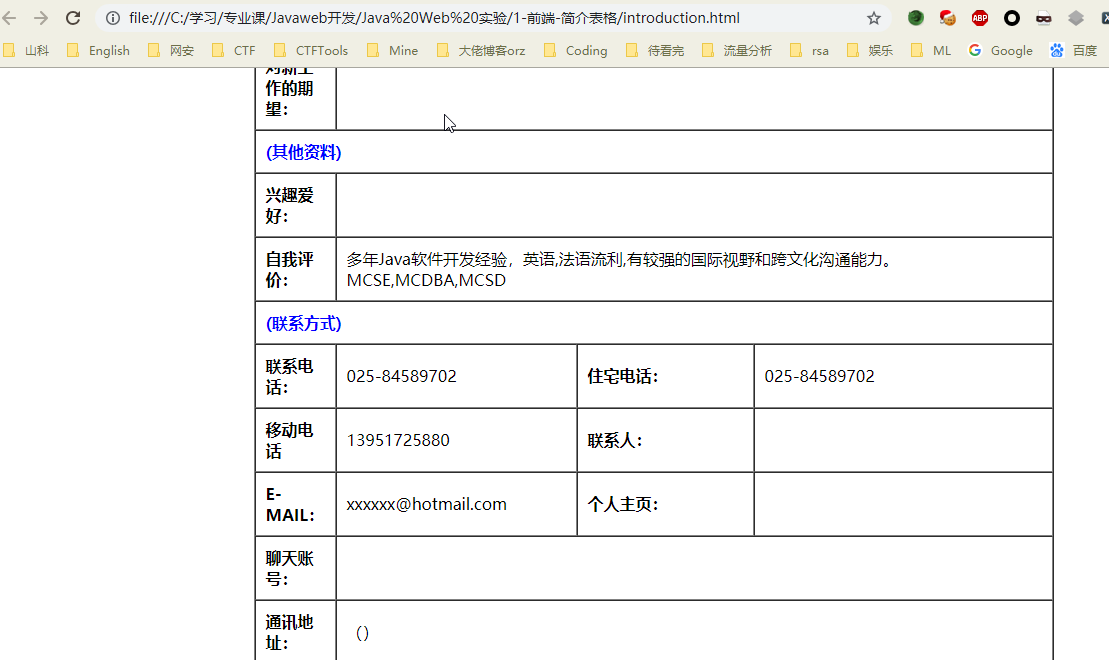
</body>

</html>

运行结果（部分截图）如下：







# 实验2 JSP与Servlet Web登录程序

## 1、实验目的

* 1. 通过实例开发，熟练掌握JSP语法及相关标签的应用。
  2. 熟练掌握Servlet作为控制器，实现用户登录的处理和界面跳转。

## 2、实验内容

采用JSP+Servlet+JavaBean+JDBC方式开发一个Web登录程序。

要求：数据库中建一个“用户名-密码”表，用户由页面上输入用户名和密码，单击【登录】按钮提交，程序通过JDBC访问数据库中的表来验证用户，验证通过转到主页并回显欢迎信息，否则跳转至出错页。

## 3、实验步骤

（1）建立数据库与表

在MySQL test数据库中，建立一个用户表userTable，表结构如下表所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 主键 | 自增 | 描述 |
| id | int | 是 | 增1 | ID号 |
| username | varchar(20) |  |  | 用户名 |
| password | varchar(20) |  |  | 密码 |

字段包括：id、username和password。其中id设为自动增长的int型，并设为主键。username和password都设为varchar型。表建好后，向其中录入两条数据记录。

（2）创建Java EE项目

在MyEclipse 中，选择主菜单【File】→【New】→【Web Project】，填写“Project Name”栏（为项目起名），项目命名为jsp\_servlet\_javabean\_jdbc

（3）编写JSP

本例只须写3个JSP文件：login.jsp（登录页）、main.jsp（主页）和error.jsp（出错页)

1. login.jsp

展开项目的工程目录树，右击“WebRoot”项，从弹出的菜单中选择【New】→【File】，写入文件名“login.jsp”，单击【Finish】按钮。

<%@ page language="java" pageEncoding="gb2312"%>

<html>

<head>

<title>简易留言板</title>

</head>

<body bgcolor="#E3E3E3">

<form action="mainServlet" method="post">

<table>

<caption>用户登录</caption>

<tr>

<td>用户名：</td>

<td>

<input type="text" name="username" size="20"/>

</td>

</tr>

<tr>

<td>密码:</td>

<td>

<input type="password" name="password" size="21"/>

</td>

</tr>

</table>

<input type="submit" value="登录"/>

<input type="reset" value="重置"/>

</form>

如果没注册单击<a href="">这里</a>注册！

</body>

</html>

【2】main.jsp

<%@ page language="java" pageEncoding="gb2312"%>

<html>

<head> <title>留言板信息</title></head>

<body>

<% out.print(request.getParameter("username"));%>，您好！欢迎登录留言板。

</body>

</html>

【3】error.jsp

<%@ page language="java" pageEncoding="gb2312"%>

<html>

<head> <title>出错</title></head>

<body>

登录失败！单击<a href="login.jsp">这里</a>返回

</body>

</html>

（4）构造JavaBean、创建JDBC

在项目src文件夹下建立包org.easybooks.test.model.vo，其中创建名为UserTable的Java类（JavaBean）；建立包org.easybooks.test.jdbc，在包下创建SqlSrvDBConn类（JDBC），并往项目中添加JDBC驱动包。

【1】SqlSrvDBConn.java

package org.easybooks.test.jdbc;

import java.sql.\*;

public class SqlSrvDBConn {

private Statement stmt;

private Connection conn;

ResultSet rs;

//在构造方法中创建数据库连接

public SqlSrvDBConn(){

stmt=null;

try{

/\*\*加载并注册 SQLServer 2008 的 JDBC 驱动\*/

//Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

//conn=DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=TEST","sa","123456");

/\*\*加载并注册 MySQL的 JDBC 驱动\*/

Class.forName("com.mysql.jdbc.Driver");

String uri = "jdbc:mysql://localhost:3306/test";

conn = DriverManager.getConnection(uri, "root", "root123");

}catch(Exception e){

e.printStackTrace();

}

rs=null;

}

//执行查询类的SQL语句，有返回集

public ResultSet executeQuery(String sql)

{

try

{

stmt=conn.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,ResultSet.CONCUR\_UPDATABLE);

rs=stmt.executeQuery(sql);

}catch(SQLException e){

System.err.println("Data.executeQuery: " + e.getMessage());

}

return rs;

}

//关闭对象

public void closeStmt()

{

try

{

stmt.close();

}catch(SQLException e){

System.err.println("Data.executeQuery: " + e.getMessage());

}

}

public void closeConn()

{

try

{

conn.close();

}catch(SQLException e){

System.err.println("Data.executeQuery: " + e.getMessage());

}

}

}

【2】UserTable.java

package org.easybooks.test.model.vo;

public class UserTable {

private Integer id;

private String username;

private String password;

public Integer getId(){

return this.id;

}

public void setId(Integer id){

this.id=id;

}

public String getUsername(){

return this.username;

}

public void setUsername(String username){

this.username=username;

}

public String getPassword(){

return this.password;

}

public void setPassword(String password){

this.password=password;

}

}

（5）编写Servlet

package org.easybooks.test.servlet;

import java.sql.\*;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import org.easybooks.test.jdbc.SqlSrvDBConn;

import org.easybooks.test.model.vo.UserTable;;

public class MainServlet extends HttpServlet{

public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{

request.setCharacterEncoding("gb2312"); //设置请求编码

String usr=request.getParameter("username"); //获取提交的用户名

String pwd=request.getParameter("password"); //获取提交的密码

boolean validated=false; //验证成功标识

SqlSrvDBConn sqlsrvdb=new SqlSrvDBConn();

HttpSession session=request.getSession(); //获得会话对象，用来保存当前登录用户的信息

UserTable user=null;

//先获得UserTable对象，如果是第一次访问该页，用户对象肯定为空，但如果是第二次甚至是第三次，

//就直接登录主页而无须再次重复验证该用户的信息

user=(UserTable)session.getAttribute("user");

//如果用户是第一次进入，会话中尚未存储user持久化对象，故为null

if(user==null){

//查询userTable表中的记录

String sql="select \* from userTable";

ResultSet rs=sqlsrvdb.executeQuery(sql); //取得结果集

try {

while(rs.next()) {

if((rs.getString("username").trim().compareTo(usr)==0)&&(rs.getString("password"). compareTo(pwd)==0)){

user=new UserTable(); //创建持久化的JavaBean对象user

user.setId(rs.getInt(1));

user.setUsername(rs.getString(2));

user.setPassword(rs.getString(3));

session.setAttribute("user", user); //把user对象存储在会话中

validated=true; //标识为true表示验证成功通过

}

}

rs.close();

} catch (SQLException e) {

e.printStackTrace();

}

sqlsrvdb.closeStmt();

sqlsrvdb.closeConn();

}

else{

validated=true; //该用户在之前已登录过并成功验证，故标识为true表示无须再验了

}

if(validated)

{

//验证成功跳转到main.jsp

response.sendRedirect("main.jsp");

}

else{

//验证失败跳转到error.jsp

response.sendRedirect("error.jsp");

}

}

public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,

IOException{

doGet(request,response);

}

}

（6）配置Servlet

Servlet编写完成后，必须在项目web.xml中进行配置方可使用。

修改项目web.xml <?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd" id="WebApp\_ID" version="3.1">

<display-name>jsp\_servlet\_javabean\_jdbc</display-name>

<welcome-file-list>

<welcome-file>login.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>mainServlet</servlet-name>

<servlet-class>org.easybooks.test.servlet.MainServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>mainServlet</servlet-name>

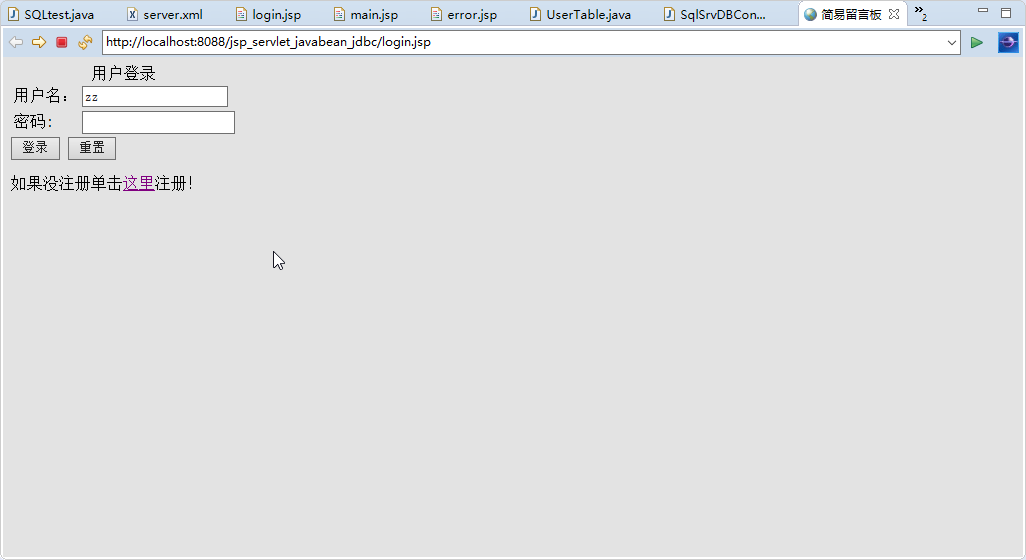
<url-pattern>/mainServlet</url-pattern>

</servlet-mapping>

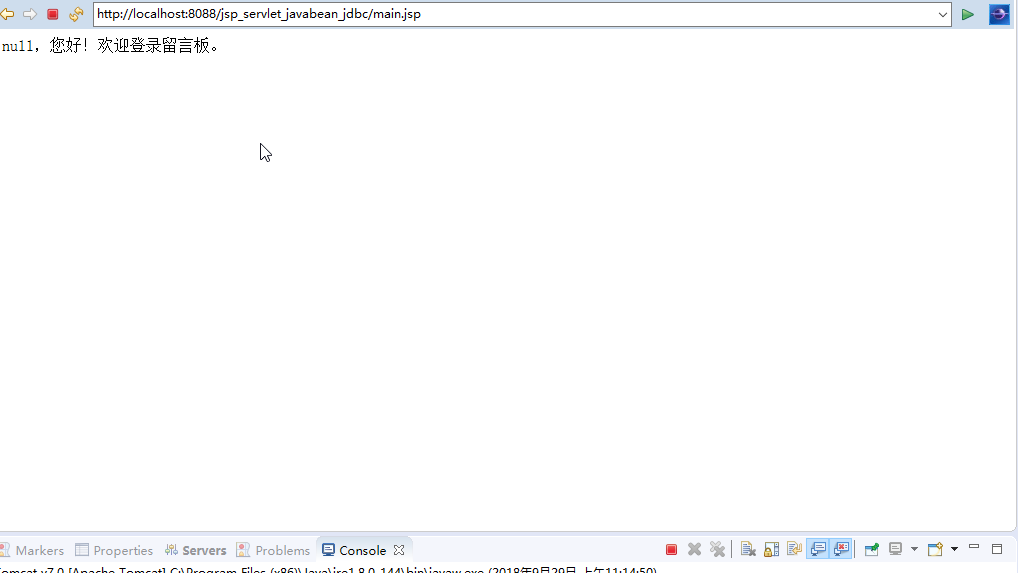
</web-app>

实验结果如下：

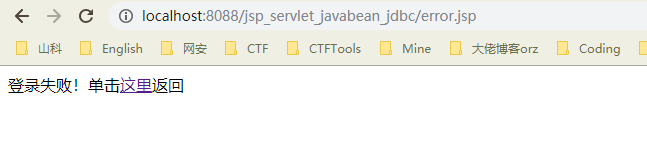
login.jsp



输入的用户名密码与数据库中一致，则跳转至main.jsp



不一致，则跳转到error.jsp



# 实验3 SpringMVC 实验

## 1、实验目的

通过实验，理解SpringMVC的工作原理和机制，理解Dispatcher和Controller的功能。

## 2、实验内容

通过SpringMVC的实例，使用SpringMVC进行请求的分发控制。

## 3、实验内容

3.1 spring框架下载：

spring-framework-3.1.1.RELEASE-with-docs.zip

3.2依赖jar包：

1、Spring框架jar包：

为了简单，将spring-framework-3.1.1.RELEASE-with-docs.zip/dist/下的所有jar包拷贝到项目的WEB-INF/lib目录下；

2、Spring框架依赖的jar包：

需要添加Apache commons logging日志，此处使用的是commons.logging-1.1.1.jar；

需要添加jstl标签库支持，此处使用的是jstl-1.1.2.jar和standard-1.1.2.jar；

3.3 SpringMVC

Spring MVC 框架是围绕一个 DispatcherServlet 来设计的，这个 Servlet 会把请求分发给各个处理器，并支持可配置的处理器映射、视图渲染、本地化、时区与主题渲染等，甚至还能支持文件上传。处理器是你的应用中注解了 @Controller 和 @RequestMapping 的类和方法，Spring 为处理器方法提供了极其多样灵活的配置。

3.4 web.xml

在/WebContent/WEB-INF/下新建web.xml文件，接下来在这个文件中配置 DispatcherServlet。

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

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"* id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>springmvc</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>springMVC</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>springMVC</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

3.5 springMVC-servlet.xml

接下来我们需要在/WebContent/WEB-INF/目录下创建一个springMVC-servlet.xml的Spring配置文件。在SpringMVC 的配置文件中告诉Spring 该到哪里去找标记为@Controller 的Controller 控制器。

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

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:task=*"http://www.springframework.org/schema/task"*

xmlns:util=*"http://www.springframework.org/schema/util"* xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-4.2.xsd*

*http://www.springframework.org/schema/util*

*http://www.springframework.org/schema/util/spring-util-4.2.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-4.2.xsd*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc-4.2.xsd*

*http://www.springframework.org/schema/task*

*http://www.springframework.org/schema/task/spring-task.xsd "*>

<context:component-scan base-package=*"cn.edu.sdust.controller"* />

<!-- 使用默认的注解映射 -->

<mvc:annotation-driven />

<mvc:resources location=*"/"* mapping=*"/index.html"* />

<bean

class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"*>

<value>/WEB-INF/jsp/</value>

</property>

<property name=*"suffix"*>

<value>.jsp</value>

</property>

</bean>

</beans>

3.6 Controller

创建一个名为springMVC的动态Web项目，并在创建的项目中的src文件夹下创建一个包cn.sdust.controller.在cn.sdust.controller下创建TestController.java文件

package cn.edu.sdust.controller;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

@Controller

@RequestMapping("/hello")

public class TestController {

@RequestMapping(method=RequestMethod.GET)

// public String hello(){

// return "hello";

// }

// @RequestMapping("h1")

public String hello1(ModelMap model){

model.addAttribute("message","hello, welcome !");

return "hello1";

}

}

@Controller 定义一个 Controller 控制器，分发处理器将会扫描使用了该注解的类的方法，并检测该方法是否使用了@RequestMapping 注解。@Controller 只是定义了一个控制器类，而使用@RequestMapping 注解的方法才是真正处理请求的处理器

3.7 hello.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

</head>

<body>

hello1 jsp

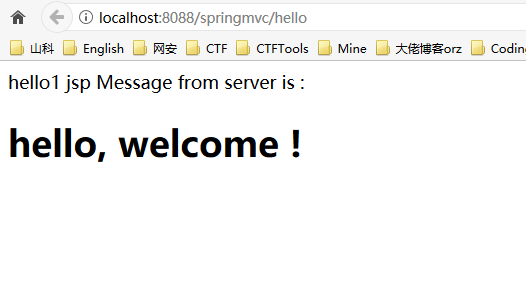
Message from server is :

<h1>${message}</h1>

</body>

</html>

3.8 实验结果



实验说明SpringMVC的工作原理：首先，根据请求去找哪个servlet，再找request mapping，再找到相应的Controller和相应的处理类，处理完毕后，返回给用户一个视图。

# 实验4 mybatis实验

## 1、实验目的

通过实例理解mybatis的运行原理和使用方法。

## 2、实验内容

利用mybatis，实现一个表的增加记录和查找操作。

## 3、实验步骤

*reference：https://www.cnblogs.com/xdp-gacl/p/4261895.html*

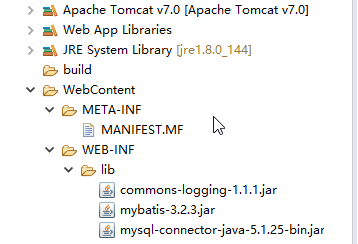
3.1. 添加相应的jar包

　　【mybatis】

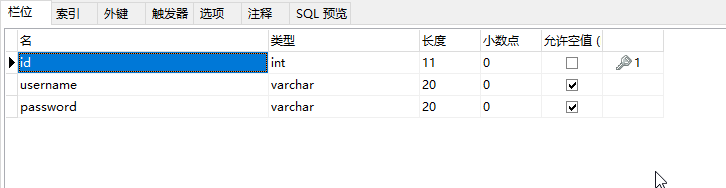
　　mybatis-3.2.3.jar

　　【MYSQL驱动包】

　　　　mysql-connector-java-5.1.25-bin.jar

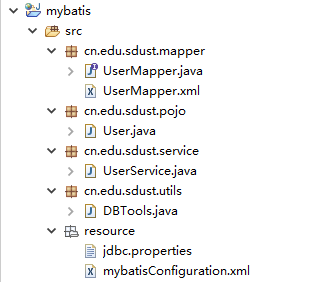


3.2 创建数据库（test）和表（usertable），针对MySQL数据库



3.3 添加Mybatis的配置文件

在src目录下创建一个mybatisConfiguration.xml文件，这是mybatis的配置文件（包括映射，数据库信息等），如下图所示

（图mybatis）

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

<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>

<!-- 引入外部配置文件 -->

<properties resource=*"resource/jdbc.properties"*></properties>

<!-- 配置mybatis运行环境 -->

<environments default=*"cybatis"*>

<environment id=*"cybatis"*>

<!-- type="JDBC" 代表使用JDBC的提交和回滚来管理事务 -->

<transactionManager type=*"JDBC"* />

<!-- mybatis提供了3种数据源类型，分别是：POOLED,UNPOOLED,JNDI -->

<!-- POOLED 表示支持JDBC数据源连接池 -->

<!-- UNPOOLED 表示不支持数据源连接池 -->

<!-- JNDI 表示支持外部数据源连接池 -->

<dataSource type=*"POOLED"*>

<property name=*"driver"* value=*"${driver}"* />

<property name=*"url"* value=*"${url}"* />

<property name=*"username"* value=*"${username}"* />

<property name=*"password"* value=*"${password}"* />

</dataSource>

</environment>

</environments>

<mappers>

<package name=*"cn/edu/sdust/mapper"*></package>

</mappers>

</configuration>

3.4、定义表所对应的实体类，如上文图mybatis所示，这个实体类User是数据库的抽象类，用于操作数据。（提供get和set方法）

package cn.edu.sdust.pojo;

public class User {

private int id;

private String username;

private String password;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

@Override

public String toString() {

// TODO Auto-generated method stub

return "User [id=" + id + ", username=" + username + ", password="+ password + "]";

}

}

3.5 定义操作users表的sql映射文件userMapper.xml创建一个mapper包，专门用于存放sql映射文件，在包中创建一个userMapper.xml文件，如前文图mybatis所示。(这个映射文件，建立了数据库和User之间的联系，主要是SQL语句)

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

<!DOCTYPE mapper PUBLIC "-//mybatis.org/DTD Mapper 3.0" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace=*"cn.edu.sdust.mapper.UserMapper"*>

<!-- 自定义返回结果集 -->

<resultMap id=*"userMap"* type=*"cn.edu.sdust.pojo.User"*>

<id property=*"id"* column=*"id"* javaType=*"java.lang.Integer"*></id>

<result property=*"username"* column=*"username"* javaType=*"java.lang.String"*></result>

<result property=*"password"* column=*"password"* javaType=*"java.lang.String"*></result>

</resultMap>

<!-- 在各种标签中的id属性必须和接口中的方法名相同 ， id属性值必须是唯一的，不能够重复使用。parameterType属性指明查询时使用的参数类型，resultType属性指明查询返回的结果集类型-->

<!-- useGeneratedKeys：（ 仅 对 insert 有 用 ） 这 会 告 诉 MyBatis 使 用 JDBC 的getGeneratedKeys

方法来取出由数据（比如：像 MySQL 和 SQLServer 这样的数据库管理系统的自动递增字段）内部生成的主键。默认值： false。 -->

<!--keyProperty： （仅对 insert有用）标记一个属性， MyBatis 会通过 getGeneratedKeys或者通过 insert 语句的 selectKey 子元素设置它的值。默认：不设置。 -->

<!--#{}中的内容，为占位符，当参数为某个JavaBean时，表示放置该Bean对象的属性值 -->

<insert id=*"insertUser"* useGeneratedKeys=*"true"* keyProperty=*"id"*>

insert into usertable (username,password) values (#{username},#{password})

</insert>

<update id=*"updateUser"* >

update usertable set username=#{username},password=#{password} where id=#{id}

</update>

<delete id=*"deleteUser"* parameterType=*"int"*>

delete from usertable where id=#{id}

</delete>

<select id=*"selectUserById"* parameterType=*"int"* resultMap=*"userMap"*>

select \* from usertable where id=#{id}

</select>

<select id=*"selectAllUser"* resultMap=*"userMap"*>

select \* from usertable

</select>

</mapper>

3.6、编写测试代码UserService类：执行定义的select语句和insert into语句

package cn.edu.sdust.service;

import java.util.\*;

import org.apache.ibatis.session.SqlSession;

import cn.edu.sdust.mapper.UserMapper;

import cn.edu.sdust.pojo.User;

import cn.edu.sdust.utils.DBTools;

public class UserService {

public static void main(String[] args){

insertUser();

selectAllUser();

selectUserById(1);

}

// 插入一个User信息

private static void insertUser(){

SqlSession session = DBTools.getSession();

UserMapper mapper = session.getMapper(UserMapper.class);

User user = new User();

user.setUsername("test111");

try{

mapper.insertUser(user);

session.commit();

}catch(Exception e){

session.rollback();

}

}

// 查询所有User信息

private static void selectAllUser(){

SqlSession session = DBTools.getSession();

UserMapper mapper = session.getMapper(UserMapper.class);

List<User> userlist = new ArrayList<User>();

try{

userlist = mapper.selectAllUser();

System.out.println("ok.");

session.commit();

}catch(Exception e){

session.rollback();

}

for (User user : userlist) {

System.out.println(user);

}

}

// 按照ID查询User信息

private static void selectUserById(int id){

SqlSession session = DBTools.getSession();

UserMapper mapper = session.getMapper(UserMapper.class);

User user = new User();

try{

user = mapper.selectUserById(id);

System.out.println("ok.");

session.commit();

}catch(Exception e){

session.rollback();

}

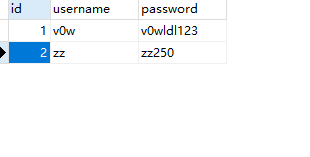
System.out.println(user);

}

}

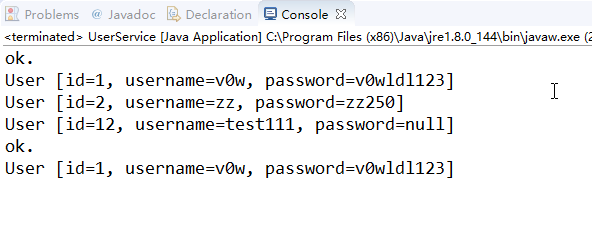
3.7 运行结果

运行前，表是这样的



运行后，是这样的（执行了插入语句和查询语句）





# 实验 5 bootdo生成代码实验

## 1、实验目的

学会利用bootdo生成代码并应用在实际项目中。

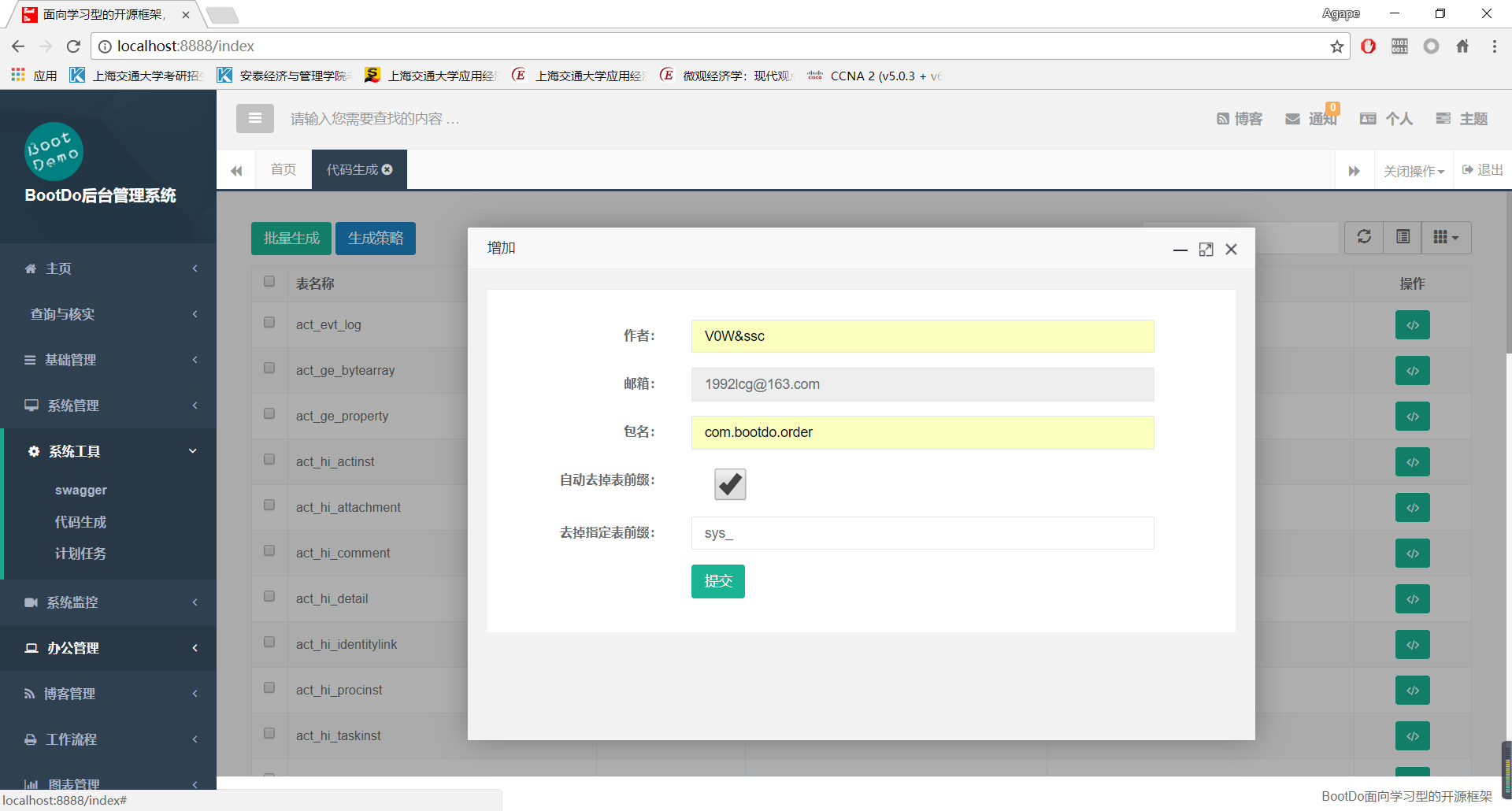
## 2、实验内容

使用bootdo，生成一个项目代码。

## 3、实验步骤

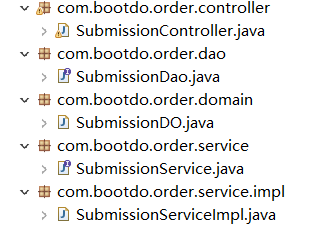
1. 配置生成策略

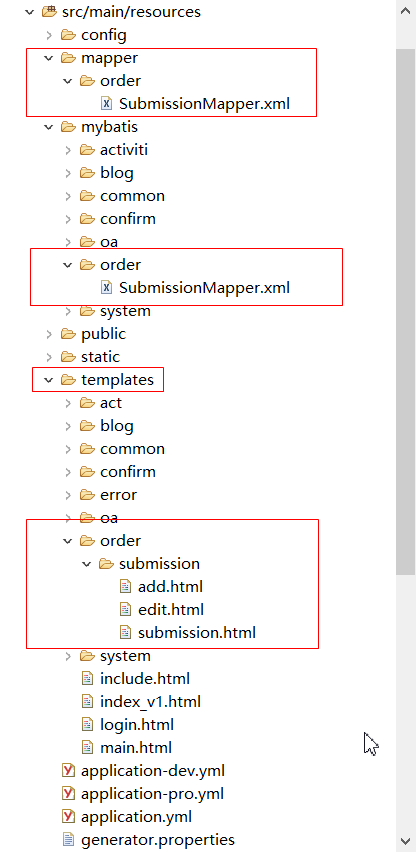
打开代码生成模块，点击生成策略，指定作者、邮箱、包名等代码注释信息



1. 生成代码并复制到项目中

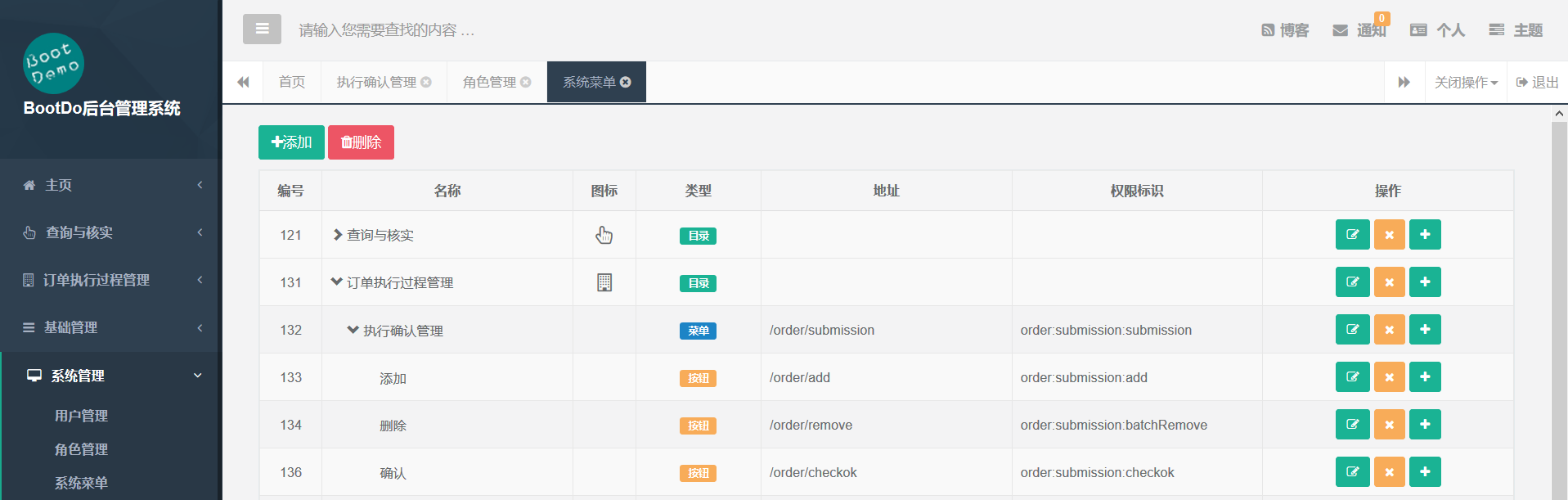






1. 设置菜单和权限

打开系统管理-系统菜单模块，设置菜单按钮等信息



打开系统管理-角色管理模块，给指定角色添加打开该模块的权限

