

实验一 Servlet 与 JSP 技术——第一个用户登录模块

一、基础实验——Servlet 与 JSP 基础开发

（一）实验目的

- 1、掌握 HttpServlet 的概念，相关 API 以及开发步骤；
- 2、掌握 JSP 技术的基本语法；
- 3、掌握 JSP 各隐含变量的使用方法；
- 4、掌握使用 Servlet 和 JSP 集成开发简单用户登录功能。

（二）基本知识与原理

- 1、Servlet 是用于实现 Web 应用程序设计的 Java 技术解决方案，旨在扩展 Web 服务器的能力，它是由 Servlet 容器（例如：Tomcat）创建并管理。
- 2、JSP（Java Server Pages）页面是包含 Java 代码和 HTML 标签的 Web 页面。它由 JSP 标签和 HTML 标签混合而成的 Web 页面，主要用于进行用户交互。

（三）实验内容及步骤

- 1、下载并解压安装 Eclipse 的 Java EE 集成开发环境，如图 1-1 所示：



图 1-1 下载并解压安装 Eclipse 的 Java EE 集成开发环境

- 2、在 Eclipse 中新建动态 Web 工程(Dynamic Web Project) javaweb-prj1，如图 1-2 所示：

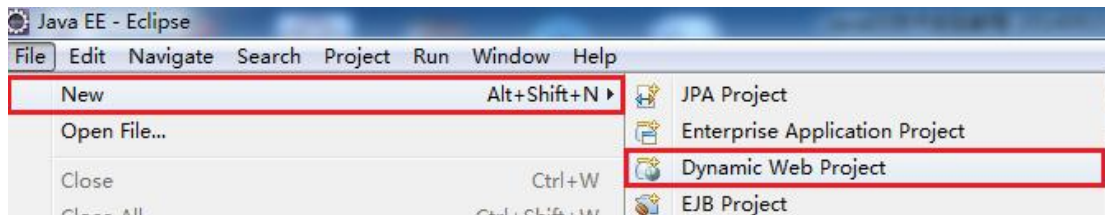


图 1-2 创建动态 Web 工程

3、为工程 javaweb-prj1 添加 Servlet 开发的外部库文件 servlet-api.jar，操作步骤如下：

- (1) 右键单击工程 javaweb-prj1 图标，在弹出的菜单中选择 **[Build Path]->[Configure Build Path]**，如图 1-3 所示；
- (2) 在弹出的窗口中选择 **[Java Build Path]** 的 **[Libraries]** 选项卡，点击按钮 **[Add External JARS]**，接着在弹出的窗口中切换到 **Tomcat 安装目录**（例如：**C:\Program Files (x86)\Apache Software Foundation\Tomcat 9.0**）下的 **lib** 目录，然后选中并打开加载 **servlet-api.jar** 库文件，最后点击按钮 **[OK]**，如图 1-4 所示；

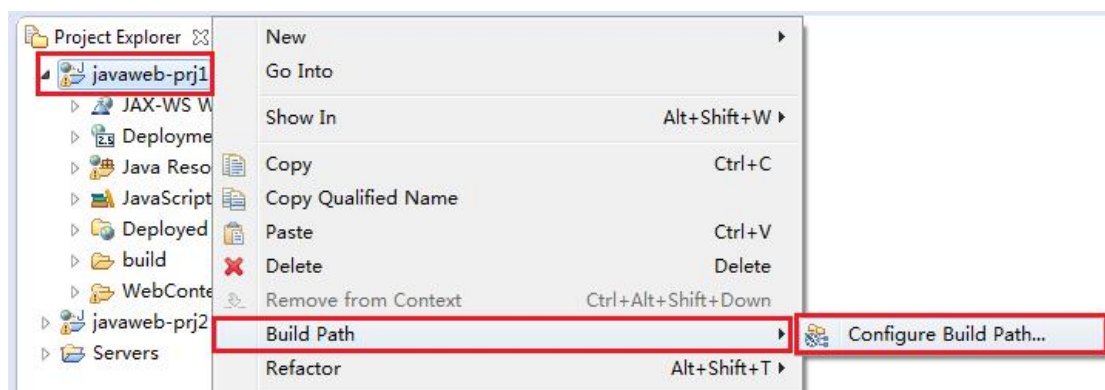


图 1-3 为工程添加外部库文件的菜单操作

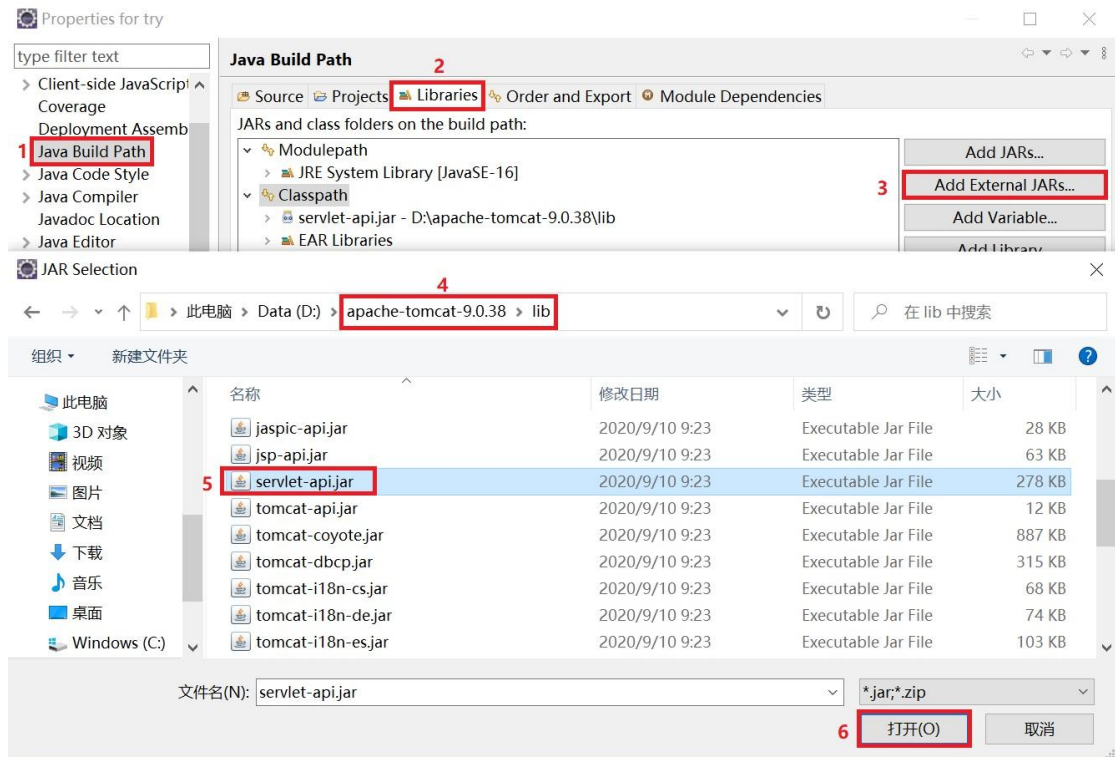


图 1-4 选择并添加外部库文件 Servlet-api.jar

- 4、在 javaweb-prj1 中，右键单击目录[WebContent]，新建用户登录页面 login.jsp，如图 1-5 所示，具体代码如下所示（注意表单 form 属性 **action** 的值，用户名与密码输入框中属性 **name** 的值）；

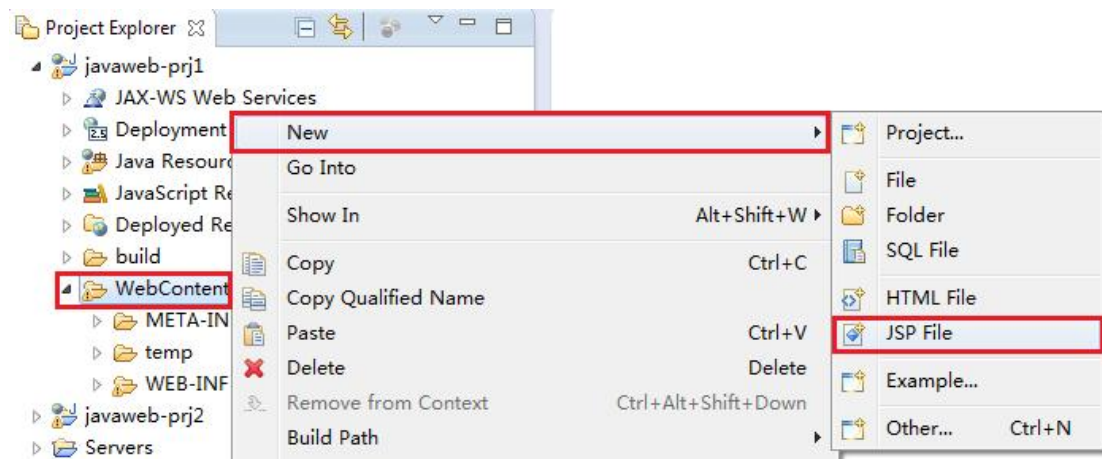


图 1-5 创建 JSP 页面

```

<%@ page language="java" contentType="text/html; charset=GBK"
    pageEncoding="GBK"%>
<!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=GBK">
<title>用户登录页面</title>
</head>
<body>
<form action="login" method="post">
    请输入用户名: <input name="username" type="text"><br>
    请输入密码: <input name="password" type="password">
        <input type="submit" value="登录">
</form>
</body>
</html>

```

- 5、在 javaweb-prj1 中，右键单击[Java Resources]->[src]目录，新建一个名称为 cn.edu.zjut 的包，然后在该包下创建一个 Servlet: LoginController.java（如图 1-6 所示），实现如下功能：用于接收 login.jsp 页面提交的用户名和密码，如果用户名和密码均为 zjut，则输出“登录成功，欢迎您！”，否则输出“用户名或密码错误！”，具体代码如下所示：

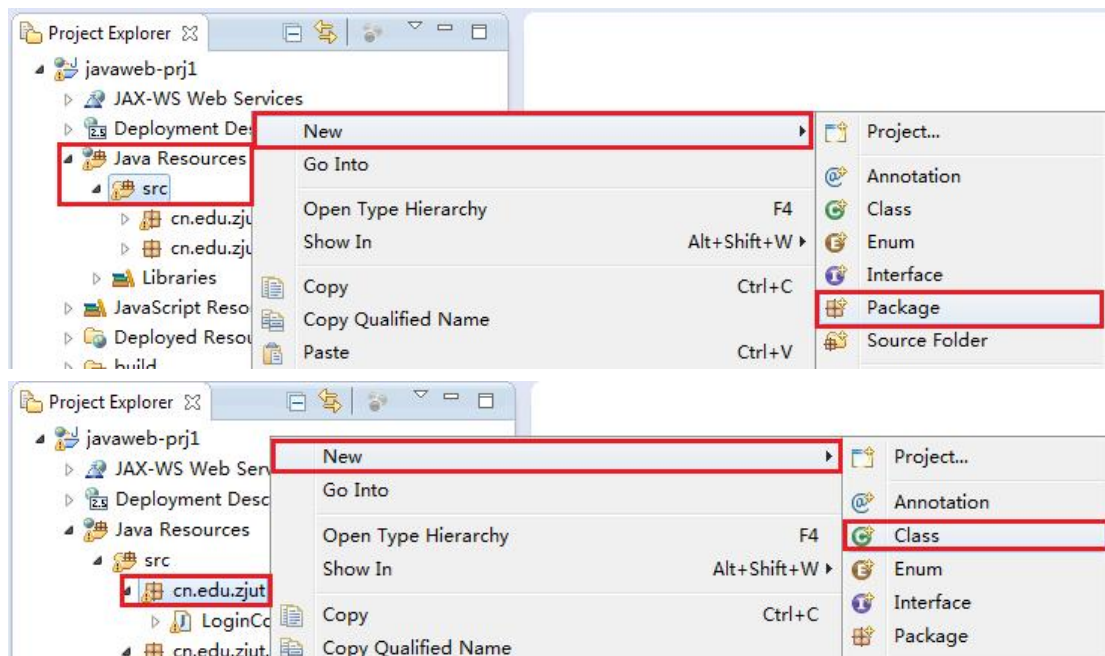


图 1-6 创建 Package 包及 Java 类

```

package cn.edu.zjut;
import java.io.*;
import javax.servlet.*;

```

```

import javax.servlet.http.*;
public class LoginController extends HttpServlet {
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=utf-8");
        PrintWriter out=response.getWriter();
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        if("zjut".equals(username) && "zjut".equals(password)){
            out.println("登录成功, 欢迎您! ");
        }else{
            out.println("用户名或密码错误! ");
        }
    }
}

```

6、在 javaweb-prj1 的 [WebContent]->[WEB-INF] 目录下新建 web.xml 文件，为 LoginController 配置 url-pattern 映射，具体代码如下所示：

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app>
    <display-name>javaweb-prj1</display-name>
    <servlet>
        <servlet-name>LoginController</servlet-name>
        <display-name>LoginController</display-name>
        <description></description>
        <servlet-class>cn.edu.zjut.LoginController</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>LoginController</servlet-name>
        <url-pattern>/login</url-pattern>
    </servlet-mapping>
</web-app>

```

7、将 javaweb-prj1 部署在 Tomcat 服务器上，具体操作步骤如下：

- (1) 下载安装好 Tomcat（本例为版本 9.0，实际应用中可选择 9.0 以上版本）；
- (2) 打开 Eclipse，选择菜单 [Window]->[Preferences]，如图 1-7 所示；

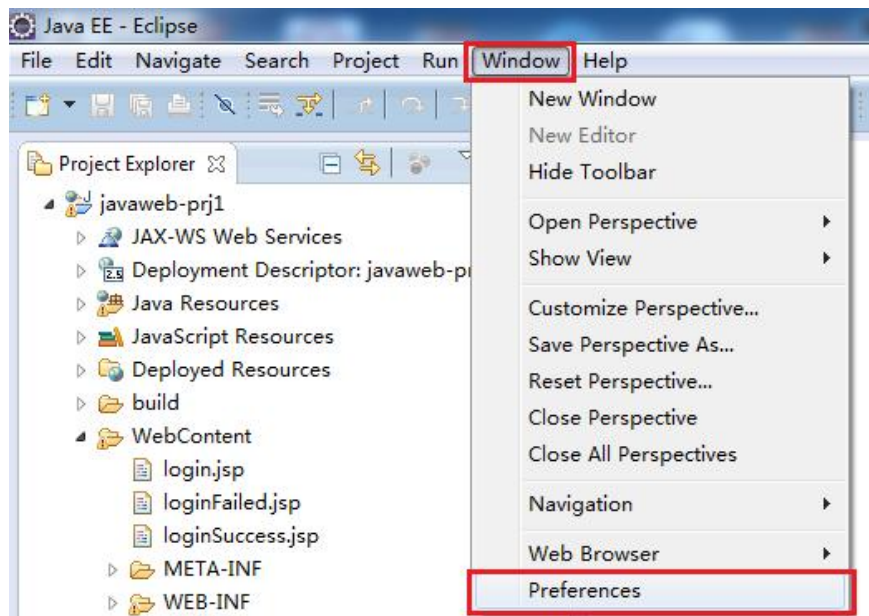


图 1-7 打开 Eclipse 的首选项

- (3) 在弹出的窗口左侧栏中选择[Server]->[Runtime Environments]，然后点击右侧栏的按钮[add]，如图 1-8 所示。

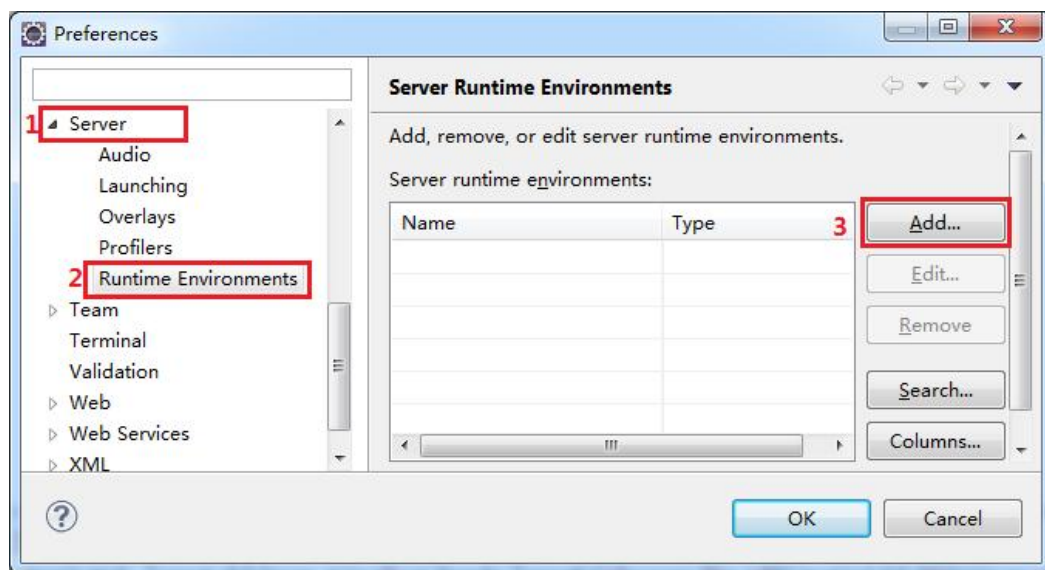


图 1-8 添加新的运行时环境

- (4) 在弹出的服务器窗口中选择[Apache Tomcat v9.0]，并点击按钮[next]，如图 1-9 所示；此时进入 Tomcat 的服务器配置窗口，点击[Browse]按钮，接着在弹出的目录选择窗口中选择 Tomcat 安装目录（例如：D:\Program Files (x86)\Apache Software Foundation\Tomcat 9.0），点击按钮[确定]完成 Tomcat 服务器的选择，然后依次点击按钮[Finish]和按钮[OK]来保存服务器配置方案，如图 1-10 所示。

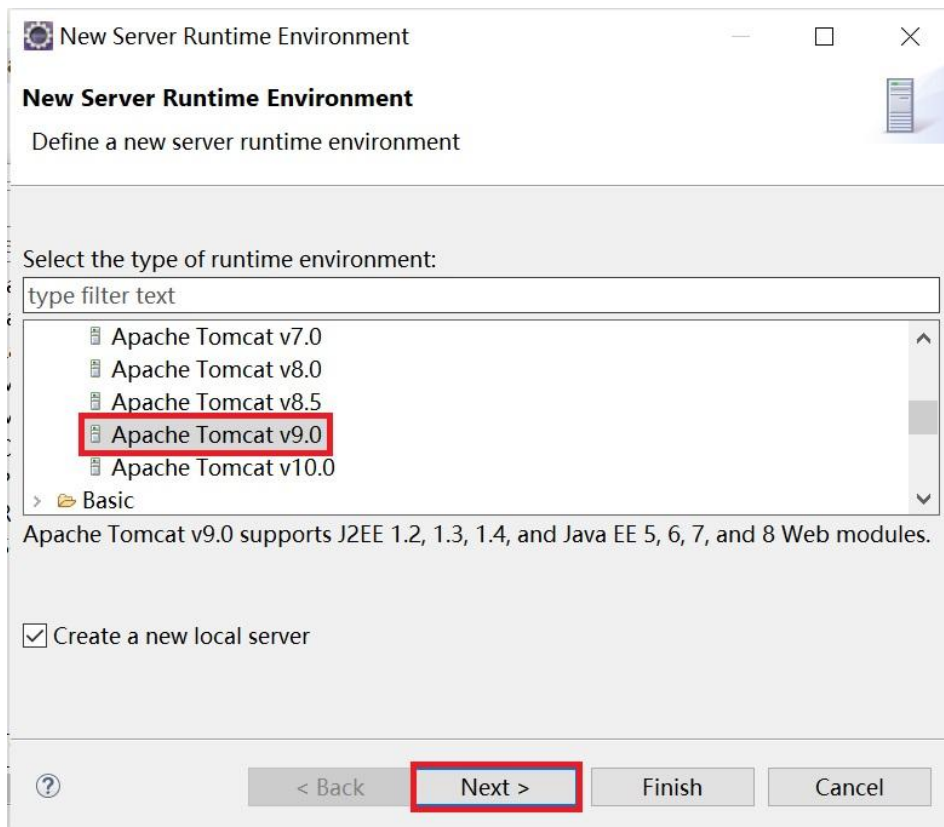


图 1-9 选定 Tomcat 服务器版本

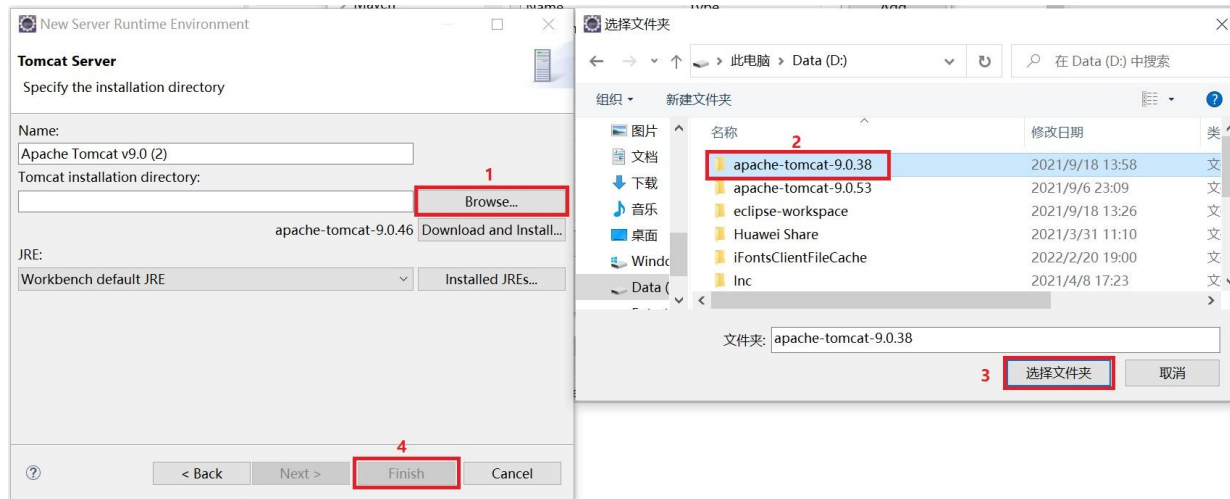


图 1-10 选择实际已安装的 Tomcat 服务器

- (5) 右键单击 `javaweb-prj1` 工程，在弹出的菜单中依次选择[Run As]->[Run on Server]，如所示；在弹出的窗口中选择刚才配置的[**Tomcat v9.0 Server**]，并点击按钮[**Finish**]，此时服务器会自动启动并加载运行 `javaweb-prj1` 工程，同时服务器启动的相关信息会出现在 **Console** 控制栏。
- (6) 打开任一浏览器，输入网址：<http://localhost:8080/javaweb-prj1/login.jsp>，即可开始访问 **login.jsp** 页面，进行相关运行调试。

- 8、运行 login.jsp 页面，输入用户名和密码并记录运行结果；
- 9、修改 login.jsp 页面，使用表格对表单域进行对齐排列，运行并观察结果；
- 10、修改 login.jsp 页面，使用 JavaScript 对用户名表单域 username 和密码表单域 password 进行校验(校验规则：不能为空且不能超过 6 位)，运行并观察结果。

(四) 实验要求

- 1、填写并上交实验报告，报告中应包括：
 - (1) 运行结果截图；
 - (2) 修改后的关键代码，及相应的运行结果或报错信息；
 - (3) 实验收获及总结。
- 2、上交程序源代码，主要代码中应加注详细的注释。

二、提高实验——Servlet 与 JSP 集成的 MVC 方案

(一) 实验目的

- 1、掌握 JavaBeans 的编写要点；
- 2、掌握 MVC 设计模式的三个组成要素：模型 Model、视图 View 和控制器 Controller；
- 3、能在 Web 应用程序设计中熟练使用 MVC 模式。

(二) 基本知识与原理

- 1、MVC 模式将交互式应用分成模型(Model)、视图(View)和控制器(Controller)三个部分。
- 2、模型(Model)是指从现实世界中抽象出来的对象模型，是应用逻辑的反映。模型封装了数据和数据的操作，是实际进行数据处理计算的地方。
- 3、视图(View)是应用和用户之间的接口，它负责将应用显现给用户和显示模型的状态。
- 4、控制器(Controller)负责视图和模型之间的交互，控制对用户输入的响应、响应方式和流程。它主要负责两方面的动作：把用户的请求分发到相应的模型；将模型的改变及时反映到视图上。
- 5、MVC 设计模式将业务逻辑和显示逻辑进行了良好的分离，使代码更加清晰，可维护性更好。

(三) 实验内容及步骤

1、在工程 javaweb-prj1 中新建用户模型 UserBean.java，具体代码如下所示：

```
package cn.edu.zjut.model;

public class UserBean {
    // 属性声明
    private String username;
    private String password;
    //构造方法
    public UserBean(){}
    // get 方法
    public String getUsername() { return username; }
    public String getPassword() { return password; }
    // set 方法
    public void setUsername(String username) {
        this.username = username;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}
```

2、在 javaweb-prj1 中新建视图 loginSuccess.jsp 和 loginFailed.jsp 页面，分别用于显示登录成功和登录失败后的页面，具体代码如下所示：

```
<!-- loginSuccess.jsp 页面源代码 -->
<%@ page language="java" contentType="text/html; charset=GBK"
    pageEncoding="GBK" import="cn.edu.zjut.model.UserBean"%>
<!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=GBK">
<title>登录成功</title>
</head>
<body>
<% UserBean user=(UserBean) (request.getAttribute("USER")); %>
登录成功,欢迎您,<%=user.getUsername() %>!
</body>
</html>
```

```
<!-- loginFailed.jsp 页面源代码 -->
<%@ page language="java" contentType="text/html; charset=GBK"
    pageEncoding="GBK"%>
```

```
<!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=GBK">
<title>登录失败</title>
</head>
<body>
登录失败，用户名或密码错误！
</body>
</html>
```

3、修改 javaweb-prj1 中的控制器 LoginController.java，实现如下功能：

- (1) 调用 UserBean 用户模型记录用户信息；
- (2) 如果用户名和密码均为 zjut，则跳转到视图 loginSuccess.jsp，否则跳转到视图 loginFailed.jsp，具体代码如下所示：

```
package cn.edu.zjut;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import cn.edu.zjut.model.*;

public class LoginController extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        UserBean user=new UserBean();
        user.setUsername(username);
        user.setPassword(password);
        if(checkUser(user)){
            request.setAttribute("USER", user);
            RequestDispatcher dispatcher = request
                .getRequestDispatcher("/loginSuccess.jsp");
            dispatcher.forward(request, response);
        }else{
            response.sendRedirect("/javaweb-prj1/loginFailed.jsp");
        }
    }
    boolean checkUser(UserBean user){
        if("zjut".equals(user.getUsername()) &&
            "zjut".equals(user.getPassword())) {
            return true;
        }else{
```

```
        return false;
    }
}
}
```

- 4、使用 javaweb-prj1 工程中的 web.xml, 为 LoginController 配置 url-pattern 映射;
- 5、将 javaweb-prj1 工程重新部署在 Tomcat 服务器上;
- 6、运行 login.jsp 页面, 输入用户名和密码并运行, 观察运行结果, 如果出现运行错误, 请尝试修正;
- 7、对 javaweb-prj1 工程做如下修改:
 - (1) 修改视图 login.jsp, 新增一个下拉框的表单域, 用于选择用户类型 (值为 “普通用户” 和 “管理员”);
 - (2) 修改用户模型 UserBean.java, 新增一个用户类型 type 属性;
 - (3) 修改控制器 LoginController.java, 要求用户名和密码均为 “zjut”, 并且用户类型为 “管理员” 才转到 loginSuccess.jsp 页面, 否则转到 loginFailed 页面。
- 8、运行并观察修改后的结果, 领会采用视图、模型和控制器分离表示逻辑和业务逻辑的优点。

(四) 实验要求

- 1、填写并上交实验报告, 报告中应包括:
 - (1) 运行结果截图;
 - (2) 根据实验过程, 查找相关资料, 总结 MVC 设计模式的优缺点;
 - (3) 实验收获及总结。
- 2、上交程序源代码, 代码中应有相关注释。

三、扩展实验——JDBC 与 DAO 设计模式

(一) 实验目的

- 1、掌握数据库操作的基本 SQL 语句;
- 2、掌握连接数据库的基本步骤;
- 3、掌握利用 JDBC 操纵数据库的基本 API 和方法;
- 4、掌握 DAO 设计模式;
- 5、掌握在一个项目中集成 MVC 设计模式和 DAO 设计模式;
- 6、进一步理解 MVC 设计模型中模型和现实中实体的对应关系;

7、理解 DAO 设计模式中业务逻辑和数据访问逻辑相分离的优点。

(二) 基本知识与原理

- 1、JDBC (Java Data Base Connectivity) 是用于执行 SQL 语句的 Java API，为多种关系数据库提供统一访问，它由一组用 Java 语言编写的类和接口组成。JDBC 提供了一种基准用于构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序。
- 2、DAO (Data Access Object) 设计模式将所有对数据源的访问操作抽象封装在一个公共 API 接口中，其中定义了应用程序中将会用到的所有事务方法。当程序开发人员需要和数据源进行交互的时候可直接使用这个接口，无需操纵数据库。

(三) 实验内容及步骤

- 1、下载和安装 PostgreSQL 数据库，下载链接地址：
<http://www.enterprisedb.com/products-services-training/pgdownload#windows>；
- 2、创建一个用户名为 dbuser，密码为 dbpassword 的数据库登录角色，然后为该角色创建一个名称为 myDB 的数据库，并在该数据库中创建一个名称为 usertable 的数据表，具体表结构如表 1-1 所示：

表 1-1 usertable 数据表

字段名称	类型	中文含义
username	Character varying(10)	登录用户名
password	Character varying(10)	登录密码
type	int	用户类型，1表示管理员，2表示普通用户。

- 3、在表 usertable 添加 3 条记录，具体如表 1-2 所示：

表 1-2 usertable 中的记录

登录用户名	密码	用户类型
zjut	zjut	1
admin	admin	1
Temp	temp	2

- 4、修改工程 javaweb-prj1：添加 PostgreSQL 驱动程序库文件到工程中（如

postgresql-42.3.4.jar) ;

- 5、修改工程 javaweb-prj1: 创建数据库操作类 UserDao.java, 实现按用户名和密码校验用户是否合法的功能:

```
package cn.edu.zjut.dao;
import java.sql.*;import javax.sql.*;import javax.naming.*;
import cn.edu.zjut.model.UserBean;
public class UserDao{
    private static final String GET_ONE_SQL =
        "SELECT * FROM usertable WHERE username=? and password=?";
    public UserDao( ){ }
    public Connection getConnection(){
        Connection conn = null;
        String driver = "org.postgresql.Driver";
        String dburl = "jdbc:postgresql://127.0.0.1:5432/myDB";
        String username = "dbuser"; //数据库登录用户名
        String password = "dbpassword"; //数据库登录密码
        try{
            Class.forName(driver); //加载数据库驱动程序
            conn = DriverManager.getConnection(dburl,username,password);
        }catch( Exception e ){ e.printStackTrace(); }
        return conn;
    }
    public boolean searchUser(UserBean user){
        // 按用户名和密码校验用户是否合法
        Connection conn = null;
        PreparedStatement pstmt = null;
        ResultSet rst=null;
        try{
            conn = getConnection();
            pstmt = conn.prepareStatement(GET_ONE_SQL);
            pstmt.setString(1, user.getUsername());
            pstmt.setString(2, user.getPassword());
            rst = pstmt.executeQuery();
            if(rst.next()){
                return true;
            }
        }catch(SQLException se){
            se.printStackTrace();
            return false;
        }finally{
            try{
                pstmt.close();
                conn.close();
            }
```

```
        }catch(SQLException se){ se.printStackTrace(); }  
    }  
    return false;  
}  
}
```

6、修改工程 javaweb-prj1 中的控制器 LoginController.java:

(1) 引入 UserDao 类，具体代码如下:

```
.....  
import cn.edu.zjut.dao.UserDAO;  
.....
```

(2) 重写 checkUser(UserBean user)方法: 实现通过数据库表 usertable 的记录来校验用户名和密码的合法性，具体修改代码如下所示:

```
.....  
boolean checkUser(UserBean user){  
    UserDao ud=new UserDao();  
    if( ud.searchUser(user) ) {  
        return true;  
    }  
    return false;  
}  
.....
```

7、将 javaweb-prj1 工程重新部署在 Tomcat 服务器上;

8、运行 login.jsp 页面，输入用户名和密码并运行，观察运行结果，如果出现运行错误，请尝试修正;

9、对 javaweb-prj1 工程进行如下修改:

- (1) 修改登录成功的条件为用户名、密码和用户类型三者匹配;
- (2) 新增一个用户注册视图 register.jsp，用于普通用户的注册;
- (3) 修改 UserDao 类，新增一个 insert(UserBean user)方法，用于插入一条记录到 usertable 表中;
- (4) 运行并观察结果。

(四) 实验要求

1、填写并上交实验报告，报告中应包括:

- (1) 运行结果截图;
- (2) 查找相关资料，总结 DAO 设计模型的优点;
- (3) 实验收获及总结。

2、上交程序源代码，代码中应有相关注释。