

第三阶段

a) 页面 jsp 动态化

- 1、在 html 页面顶行添加 page 指令。
- 2、修改文件后缀名为：.jsp
- 3、使用 IDEA 搜索替换.html 为.jsp(快捷键：Ctrl+Shift+R)

b)抽取页面中相同的内容

i. head 中 css、jquery、base 标签

```
<base href="http://localhost:8080/book/">
<link type="text/css" rel="stylesheet" href="static/css/style.css">
<script type="text/javascript" src="static/script/jquery-1.7.2.js"></script>
```

ii. 每个页面的页脚

```
<div id="bottom">
  <span>
    书城.Copyright &copy;2015
  </span>
</div>
```

iii. 登录成功后的菜单

[illegible]

iv. manager 模块的菜单

```
<div>
    <a href="book_manager.jsp">图书管理</a>
    <a href="order_manager.jsp">订单管理</a>
    <a href="../../index.jsp">返回商城</a>
</div>
```

动态获取 basePath

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

<%
    String basePath = request.getScheme()
        + "://"
        + request.getServerName()
        + ":"
        + request.getServerPort()
        + request.getContextPath()
        + "/";
%>

<base href="<%=basePath%>">
<link type="text/css" rel="stylesheet" href="static/css/style.css">
<script type="text/javascript" src="static/script/jquery-1.7.2.js"></script>
```

c) 登录，注册错误提示，及表单回显

```
package com.admiral.web;

import com.admiral.pojo.User;
import com.admiral.service.UserService;
import com.admiral.service.impl.UserServiceImpl;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
```

```

public class LoginServlet extends HttpServlet {

    private UserService userService = new UserServiceImpl();
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求参数
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        //2.调用 UserService 的 login 方法
        User login = userService.login(new User(null, username, password,
null));
        if(login!=null){
            //成功,跳转到登陆成功页面

            request.getRequestDispatcher("/pages/user/login_success.jsp").forward(request
, response);
        }else {
            //保存错误信息到request域中
            request.setAttribute("errorMsg", "用户名或密码错误!");
            request.setAttribute("username", username);
            //失败.跳转到登陆页面

            request.getRequestDispatcher("/pages/user/login.jsp").forward(request, respons
e);
        }
    }
}

```

login.jsp

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>会员登录页面</title>
    <!-- 静态包含 base css样式 jquery -->
    <%@ include file="/pages/common/header.jsp"%>
</head>
<body>
<div id="login_header">
    
</div>

<div class="login_banner">

    <div id="l_content">

```

```

        <span class="login_word">欢迎登录</span>
    </div>

    <div id="content">
        <div class="login_form">
            <div class="login_box">
                <div class="tit">
                    <h1>会员登陆</h1>
                    <a href="regist.jsp">立即注册</a>
                </div>
                <div class="msg_cont">
                    <b></b>
                    <span class="errorMsg">
<%=request.getAttribute("errorMsg")==null?"请输入用户名和密
码":request.getAttribute("errorMsg")%></span>
                </div>
                <div class="form">
                    <form action="loginServlet" method="post">
                        <label>用户名称: </label>
                        <input class="itxt" type="text" placeholder="请输入用户
名" autocomplete="off" tabindex="1"
                            name="username" value="
<%=request.getAttribute("username")==null?"":request.getAttribute("username")%
>" />
                        <br/>
                        <br/>
                        <label>用户密码: </label>
                        <input class="itxt" type="password" placeholder="请输入
密码" autocomplete="off" tabindex="1"
                            name="password" />
                        <br/>
                        <br/>
                        <input type="submit" value="登录" id="sub_btn" />
                    </form>
                </div>
            </div>
        </div>
    </div>
<!-- 静态包含页脚 --%>
<%@ include file="/pages/common/footer.jsp"%>
</body>
</html>

```

注册

```
package com.admiral.web;
```

```
import com.admiral.pojo.User;
import com.admiral.service.UserService;
import com.admiral.service.impl.UserServiceImpl;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

public class RegistServlet extends HttpServlet {

    private UserService userService = new UserServiceImpl();

    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求参数
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        String email = request.getParameter("email");
        String code = request.getParameter("code");

        //2. 坚持验证码是否正确
        if("abcde".equalsIgnoreCase(code)){//正确

            //检查用户名是否可用
            if(userService.existsUsername(username)){
                //不可用,跳转到注册页面
                System.out.println("用户名[" + username + "]已存在");

                request.setAttribute("errorMsg","用户名已存在");
                request.setAttribute("username",username);
                request.setAttribute("email",email);

                request.getRequestDispatcher("/pages/user/regist.jsp").forward(request,response);
            }else {
                //可用,调用 Service 保存用户
                userService.register(new User(null,username,password,email));
                //跳转到注册成功页面

                request.getRequestDispatcher("/pages/user/regist_success.jsp").forward(request,response);
            }
        }else {
            //验证码不正确,跳转到注册页面
            System.out.println("验证码[" + code + "]错误");
        }
    }
}
```

```

        request.setAttribute("errorMsg", "验证码错误");
        request.setAttribute("username", username);
        request.setAttribute("email", email);

        request.getRequestDispatcher("/pages/user/regist.jsp").forward(request, response);
    }

}

}

```

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>会员注册页面</title>
    <!-- 静态包含 base css样式 jquery -->
    <%@ include file="/pages/common/header.jsp"%>
    <script type="text/javascript">
        // 页面加载完成之后
        $(function () {
            // 给注册绑定单击事件
            $("#sub_btn").click(function () {
                // 验证用户名：必须由字母，数字下划线组成，并且长度为5到12位
                //1 获取用户名输入框里的内容
                var usernameText = $("#username").val();
                //2 创建正则表达式对象
                var usernamePatt = /^\\w{5,12}$/;
                //3 使用test方法验证
                if (!usernamePatt.test(usernameText)) {
                    //4 提示用户结果
                    $("#span.errorMsg").text("用户名不合法!");

                    return false;
                }

                // 验证密码：必须由字母，数字下划线组成，并且长度为5到12位
                //1 获取用户名输入框里的内容
                var passwordText = $("#password").val();
                //2 创建正则表达式对象
                var passwordPatt = /^\\w{5,12}$/;
                //3 使用test方法验证
                if (!passwordPatt.test(passwordText)) {

```

```

        //4 提示用户结果
        $("span.errorMsg").text("密码不合法! ");

        return false;
    }

    // 验证确认密码: 和密码相同
    //1 获取确认密码内容
    var repwdText = $("#repwd").val();
    //2 和密码相比较
    if (repwdText != passwordText) {
        //3 提示用户
        $("span.errorMsg").text("确认密码和密码不一致! ");

        return false;
    }

    // 邮箱验证: xxxxx@xxx.com
    //1 获取邮箱里的内容
    var emailText = $("#email").val();
    //2 创建正则表达式对象
    var emailPatt = /^[a-z\d]+(\.[a-z\d]+)*@([\da-z](-[\da-z])?)+
(\.{1,2}[a-z]+)$/;
    //3 使用test方法验证是否合法
    if (!emailPatt.test(emailText)) {
        //4 提示用户
        $("span.errorMsg").text("邮箱格式不合法! ");

        return false;
    }

    // 验证码: 现在只需要验证用户已输入。因为还没讲到服务器。验证码生成。
    var codeText = $("#code").val();

    //去掉验证码前后空格
    // alert("去空格前: [" +codeText+"]")
    codeText = $.trim(codeText);
    // alert("去空格后: [" +codeText+"]")

    if (codeText == null || codeText == "") {
        //4 提示用户
        $("span.errorMsg").text("验证码不能为空! ");

        return false;
    }

    // 去掉错误信息
    $("span.errorMsg").text("");

```

```

    });

    });

</script>
<style type="text/css">
    .login_form {
        height: 420px;
        margin-top: 25px;
    }

</style>
</head>
<body>
<div id="login_header">
    
</div>

<div class="login_banner">

    <div id="l_content">
        <span class="login_word">欢迎注册</span>
    </div>

    <div id="content">
        <div class="login_form">
            <div class="login_box">
                <div class="tit">
                    <h1>注册会员</h1>
                    <span class="errorMsg">
<%=request.getAttribute("errorMsg")==null?"":request.getAttribute("errorMsg")%>
</span>
                </div>
                <div class="form">
                    <form action="registServlet" method="post">
                        <label>用户名称: </label>
                        <input class="itxt" type="text" placeholder="请输入用户
名"
                                autocomplete="off" tabindex="1" name="username"
                                id="username"
                                value="
<%=request.getAttribute("username")==null?"":request.getAttribute("username")%>
"/>
                        <br/>
                        <br/>
                        <label>用户密码: </label>
                        <input class="itxt" type="password" placeholder="请输入
密码"

```



```

        autocomplete="off" tabindex="1" name="password"
id="password"/>

        <br/>
        <br/>
        <label>确认密码: </label>
        <input class="itxt" type="password" placeholder="确认密
码"

        autocomplete="off" tabindex="1" name="repwd"
id="repwd"/>

        <br/>
        <br/>
        <label>电子邮件: </label>
        <input class="itxt" type="text" placeholder="请输入邮箱
地址"

        autocomplete="off" tabindex="1" name="email"
id="email"

        value="
<%=request.getAttribute("email")==null?"":request.getAttribute("email")%>/>
        <br/>
        <br/>
        <label>验证码: </label>
        <input class="itxt" type="text" name="code"
style="width: 150px;" id="code"/>
        
        <br/>
        <br/>
        <input type="submit" value="注册" id="sub_btn"/>
    </form>
</div>

</div>
</div>
</div>
<!-- 静态包含页脚 -->
<%@ include file="/pages/common/footer.jsp"%>
</body>
</html>

```

d)BaseServlet 的抽取

在实际的项目开发中，一个模块，一般只使用一个 Servlet 程序。

代码优化一：代码优化：合并 LoginServlet 和 RegistServlet 程序为 UserServlet 程序

```

package com.admiral.web;

import com.admiral.pojo.User;
import com.admiral.service.UserService;
import com.admiral.service.impl.UserServiceImpl;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

public class UserServlet extends HttpServlet {
    private UserService userService = new UserServiceImpl();

    protected void login(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求参数
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        //2.调用 UserService 的 login 方法
        User login = userService.login(new User(null, username, password,
null));
        if (login != null) {
            //成功,跳转到登陆成功页面

            request.getRequestDispatcher("/pages/user/login_success.jsp").forward(request
, response);
        } else {
            //保存错误信息到request域中
            request.setAttribute("errorMsg", "用户名或密码错误!");
            request.setAttribute("username", username);
            //失败.跳转到登陆页面

            request.getRequestDispatcher("/pages/user/login.jsp").forward(request,
response);
        }
    }

    protected void regist(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求参数
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        String email = request.getParameter("email");
        String code = request.getParameter("code");

        //2. 坚持验证码是否正确
        if ("abcde".equalsIgnoreCase(code)) { //正确

```

```

        //检查用户名是否可用
        if (userService.existsUsername(username)) {
            //不可用,跳转到注册页面
            System.out.println("用户名[" + username + "]已存在");

            request.setAttribute("errorMsg", "用户名已存在");
            request.setAttribute("username", username);
            request.setAttribute("email", email);

            request.getRequestDispatcher("/pages/user/regist.jsp").forward(request,
response);
        } else {
            //可用,掉用 Service 保存用户
            userService.register(new User(null, username, password,
email));

            //跳转到注册成功页面

            request.getRequestDispatcher("/pages/user/regist_success.jsp").forward(reques
t, response);
        }

    } else {
        //验证码不正确,跳转到注册页面
        System.out.println("验证码[" + code + "]错误");

        request.setAttribute("errorMsg", "验证码错误");
        request.setAttribute("username", username);
        request.setAttribute("email", email);

        request.getRequestDispatcher("/pages/user/regist.jsp").forward(request,
response);
    }
}

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    String action = request.getParameter("action");
    System.out.println(action);
    if ("login".equals(action)) {
        login(request, response);
    } else if ("regist".equals(action)) {
        regist(request, response);
    }
}
}

```

在 login.jsp 中添加隐藏域

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>会员登录页面</title>
    <!-- 静态包含 base css样式 jquery --%>
    <%@ include file="/pages/common/header.jsp"%>
</head>
<body>
<div id="login_header">
    
</div>

<div class="login_banner">

    <div id="l_content">
        <span class="login_word">欢迎登录</span>
    </div>

    <div id="content">
        <div class="login_form">
            <div class="login_box">
                <div class="tit">
                    <h1>会员登陆</h1>
                    <a href="regist.jsp">立即注册</a>
                </div>
                <div class="msg_cont">
                    <b></b>
                    <span class="errorMsg">
<%=request.getAttribute("errorMsg")==null?"请输入用户名和密码":request.getAttribute("errorMsg")%></span>
                </div>
                <div class="form">
                    <form action="userService" method="post">
                        <input type="hidden" name="action" value="login">
                        <label>用户名称: </label>
                        <input class="itxt" type="text" placeholder="请输入用户名" autocomplete="off" tabindex="1"
                                name="username" value="
<%=request.getAttribute("username")==null?"":request.getAttribute("username")%>
                    </form>
                    <br/>
                    <br/>
                    <label>用户密码: </label>

```

```

        <input class="itxt" type="password" placeholder="请输入
密码" autocomplete="off" tabindex="1"
        name="password" />
        <br/>
        <br/>
        <input type="submit" value="登录" id="sub_btn" />
    </form>
</div>

</div>
</div>
</div>
<!-- 静态包含页脚 -->
<%@ include file="/pages/common/footer.jsp"%>
</body>
</html>

```

Regist.jsp

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>会员注册页面</title>
    <!-- 静态包含 base css样式 jquery -->
    <%@ include file="/pages/common/header.jsp"%>
    <script type="text/javascript">
        // 页面加载完成之后
        $(function () {
            // 给注册绑定单击事件
            $("#sub_btn").click(function () {
                // 验证用户名：必须由字母，数字下划线组成，并且长度为5到12位
                //1 获取用户名输入框里的内容
                var usernameText = $("#username").val();
                //2 创建正则表达式对象
                var usernamePatt = /^\\w{5,12}$/;
                //3 使用test方法验证
                if (!usernamePatt.test(usernameText)) {
                    //4 提示用户结果
                    $("span.errorMsg").text("用户名不合法!");

                    return false;
                }

                // 验证密码：必须由字母，数字下划线组成，并且长度为5到12位
            }
        });
    </script>

```

```
//1 获取用户名输入框里的内容
var passwordText = $("#password").val();
//2 创建正则表达式对象
var passwordPatt = /^w{5,12}$/;
//3 使用test方法验证
if (!passwordPatt.test(passwordText)) {
    //4 提示用户结果
    $("span.errorMsg").text("密码不合法! ");

    return false;
}

// 验证确认密码：和密码相同
//1 获取确认密码内容
var repwdText = $("#repwd").val();
//2 和密码相比较
if (repwdText != passwordText) {
    //3 提示用户
    $("span.errorMsg").text("确认密码和密码不一致! ");

    return false;
}

// 邮箱验证：xxxxx@xxx.com
//1 获取邮箱里的内容
var emailText = $("#email").val();
//2 创建正则表达式对象
var emailPatt = /^[a-z\d]+(\.[a-z\d]+)*@([\da-z](-[\da-z])?)+(\.{1,2}[a-z]+)$/;
//3 使用test方法验证是否合法
if (!emailPatt.test(emailText)) {
    //4 提示用户
    $("span.errorMsg").text("邮箱格式不合法! ");

    return false;
}

// 验证码：现在只需要验证用户已输入。因为还没讲到服务器。验证码生成。
var codeText = $("#code").val();

//去掉验证码前后空格
// alert("去空格前: ["+codeText+"]")
codeText = $.trim(codeText);
// alert("去空格后: ["+codeText+"]")

if (codeText == null || codeText == "") {
    //4 提示用户
    $("span.errorMsg").text("验证码不能为空! ");
}
```

```

        return false;
    }

    // 去掉错误信息
    $("span.errorMsg").text("");

    });

});

</script>
<style type="text/css">
    .login_form {
        height: 420px;
        margin-top: 25px;
    }

</style>
</head>
<body>
<div id="login_header">
    
</div>

<div class="login_banner">

    <div id="l_content">
        <span class="login_word">欢迎注册</span>
    </div>

    <div id="content">
        <div class="login_form">
            <div class="login_box">
                <div class="tit">
                    <h1>注册会员</h1>
                    <span class="errorMsg">
<%=request.getAttribute("errorMsg")==null?"":request.getAttribute("errorMsg")%>
</span>
                </div>
                <div class="form">
                    <form action="userServlet" method="post">
                        <input type="hidden" name="action" value="regist">
                        <label>用户名称: </label>
                        <input class="itxt" type="text" placeholder="请输入用户
名"
                                autocomplete="off" tabindex="1" name="username"
                                id="username"

```

```

        value="
<%=request.getAttribute("username")==null?"":request.getAttribute("username")%
>"/>

        <br/>
        <br/>
        <label>用户密码: </label>
        <input class="itxt" type="password" placeholder="请输入
密码"

        autocomplete="off" tabindex="1" name="password"
id="password"/>

        <br/>
        <br/>
        <label>确认密码: </label>
        <input class="itxt" type="password" placeholder="确认密
码"

        autocomplete="off" tabindex="1" name="repwd"
id="repwd"/>

        <br/>
        <br/>
        <label>电子邮件: </label>
        <input class="itxt" type="text" placeholder="请输入邮箱
地址"

        autocomplete="off" tabindex="1" name="email"
id="email"

        value="
<%=request.getAttribute("email")==null?"":request.getAttribute("email")%>"/>
        <br/>
        <br/>
        <label>验证码: </label>
        <input class="itxt" type="text" name="code"
style="width: 150px;" id="code"/>
        
        <br/>
        <br/>
        <input type="submit" value="注册" id="sub_btn"/>
    </form>
</div>

</div>
</div>
</div>
<!-- 静态包含页脚 -->
<%@ include file="/pages/common/footer.jsp"%>
</body>
</html>

```


优化代码二：使用反射优化大量 else if 代码

```
String action = request.getParameter("action");
System.out.println(action);

try {
    Method declaredMethod = this.getClass().getDeclaredMethod(action,
HttpServletRequest.class, HttpServletResponse.class);
    declaredMethod.invoke(this, request, response);
} catch (Exception e) {
    e.printStackTrace();
}
```

UserServlet.java 完整代码

```
package com.admiral.web;

import com.admiral.pojo.User;
import com.admiral.service.UserService;
import com.admiral.service.impl.UserServiceImpl;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.lang.reflect.Method;

public class UserServlet extends HttpServlet {
    private UserService userService = new UserServiceImpl();

    protected void login(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //1.获取请求参数
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        //2.调用 UserService 的 login 方法
        User login = userService.login(new User(null, username, password,
null));
        if (login != null) {
            //成功,跳转到登陆成功页面

            request.getRequestDispatcher("/pages/user/login_success.jsp").forward(request
, response);
        } else {
            //保存错误信息到request域中
        }
    }
}
```

```

        request.setAttribute("errorMsg", "用户名或密码错误!");
        request.setAttribute("username", username);
        //失败.跳转到登陆页面

    request.getRequestDispatcher("/pages/user/login.jsp").forward(request,
response);
    }
}

protected void regist(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    //1.获取请求参数
    String username = request.getParameter("username");
    String password = request.getParameter("password");
    String email = request.getParameter("email");
    String code = request.getParameter("code");

    //2. 坚持验证码是否正确
    if ("abcde".equalsIgnoreCase(code)) { //正确

        //检查用户名是否可用
        if (userService.existsUsername(username)) {
            //不可用,跳转到注册页面
            System.out.println("用户名[" + username + "]已存在");

            request.setAttribute("errorMsg", "用户名已存在");
            request.setAttribute("username", username);
            request.setAttribute("email", email);

            request.getRequestDispatcher("/pages/user/regist.jsp").forward(request,
response);
        } else {
            //可用,调用 Service 保存用户
            userService.register(new User(null, username, password,
email));

            //跳转到注册成功页面

            request.getRequestDispatcher("/pages/user/regist_success.jsp").forward(request,
response);
        }
    } else {
        //验证码不正确,跳转到注册页面
        System.out.println("验证码[" + code + "]错误");

        request.setAttribute("errorMsg", "验证码错误");
        request.setAttribute("username", username);
        request.setAttribute("email", email);
    }
}

```

```

        request.getRequestDispatcher("/pages/user/regist.jsp").forward(request,
response);
    }
}

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    String action = request.getParameter("action");
    System.out.println(action);

    try {
        Method declaredMethod = this.getClass().getDeclaredMethod(action,
HttpServletRequest.class, HttpServletResponse.class);
        declaredMethod.invoke(this, request, response);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

代码优化三：抽取 BaseServlet 程序。

```

package com.admiral.web;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.lang.reflect.Method;

/**
 * @author 白世鑫
 * @title: BaseServlet
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/1 4:50 上午
 */
public abstract class BaseServlet extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        String action = request.getParameter("action");
        System.out.println(action);
    }
}

```

```

        try {
            Method declaredMethod = this.getClass().getDeclaredMethod(action,
                HttpServletRequest.class, HttpServletResponse.class);
            declaredMethod.invoke(this, request, response);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

修改 UserServlet 继承 BaseServlet

```

package com.admiral.web;

import com.admiral.pojo.User;
import com.admiral.service.UserService;
import com.admiral.service.impl.UserServiceImpl;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

public class UserServlet extends BaseServlet {
    private UserService userService = new UserServiceImpl();

    protected void login(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException {
        //1.获取请求参数
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        //2.调用 UserService 的 login 方法
        User login = userService.login(new User(null, username, password,
            null));
        if (login != null) {
            //成功,跳转到登陆成功页面

            request.getRequestDispatcher("/pages/user/login_success.jsp").forward(request
                , response);
        } else {
            //保存错误信息到request域中
            request.setAttribute("errorMsg", "用户名或密码错误!");
            request.setAttribute("username", username);
            //失败.跳转到登陆页面

            request.getRequestDispatcher("/pages/user/login.jsp").forward(request,
                response);
        }
    }
}

```

```

    }
}

protected void regist(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    //1.获取请求参数
    String username = request.getParameter("username");
    String password = request.getParameter("password");
    String email = request.getParameter("email");
    String code = request.getParameter("code");

    //2. 坚持验证码是否正确
    if ("abcde".equalsIgnoreCase(code)) { //正确

        //检查用户名是否可用
        if (userService.existsUsername(username)) {
            //不可用,跳转到注册页面
            System.out.println("用户名[" + username + "]已存在");

            request.setAttribute("errorMsg", "用户名已存在");
            request.setAttribute("username", username);
            request.setAttribute("email", email);

            request.getRequestDispatcher("/pages/user/regist.jsp").forward(request,
response);
        } else {
            //可用,掉用 Service 保存用户
            userService.register(new User(null, username, password,
email));

            //跳转到注册成功页面

            request.getRequestDispatcher("/pages/user/regist_success.jsp").forward(reques
t, response);
        }
    } else {
        //验证码不正确,跳转到注册页面
        System.out.println("验证码[" + code + "]错误");

        request.setAttribute("errorMsg", "验证码错误");
        request.setAttribute("username", username);
        request.setAttribute("email", email);

        request.getRequestDispatcher("/pages/user/regist.jsp").forward(request,
response);
    }
}

```

```
}
```

e)数据的封装和抽取 BeanUtils 的使用

BeanUtils 工具类，它可以一次性的把所有请求的参数注入到 JavaBean 中。

BeanUtils 工具类，经常用于把 Map 中的值注入到 JavaBean 中，或者是对象属性值的拷贝操作。

BeanUtils 它不是 Jdk 的类。而是第三方的工具类。所以需要导包。

- 1、导入需要的 jar 包：
 - commons-beanutils-1.8.0.jar
 - commons-logging-1.1.1.jar
- 2、编写 WebUtils 工具类使用：

```
package com.admiral.utils;

import org.apache.commons.beanutils.BeanUtils;

import javax.servlet.http.HttpServletRequest;

/**
 * @author 白世鑫
 * @title: BeanUtils
 * @projectName JavaWeb
 * @description:
 * @date 2020/9/1 5:15 上午
 */
public class WebUtils {

    public static <T>T copyParamToBean(HttpServletRequest request, T bean){
        try {
            BeanUtils.populate(bean,request.getParameterMap());
            // System.out.println("注入之后:" + bean);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return bean;
    }
}
```

第四阶段 使用 EL 表达式修改表单回显

以登录为示例：

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>会员登录页面</title>

    <!-- 静态包含 base css样式 JQuery -->
    <%@ include file="/pages/common/header.jsp"%>
</head>
<body>
<div id="login_header">
    
</div>

<div class="login_banner">

    <div id="l_content">
        <span class="login_word">欢迎登录</span>
    </div>

    <div id="content">
        <div class="login_form">
            <div class="login_box">
                <div class="tit">
                    <h1>会员登陆</h1>
                    <a href="regist.jsp">立即注册</a>
                </div>
                <div class="msg_cont">
                    <b></b>
                </div>
            </div>
            <div class="errorMsg">
                <%=request.getAttribute("errorMsg")==null?"请输入用户名和密码":request.getAttribute("errorMsg")%></div>
            <div class="errorMsg">
                <${empty requestScope.errorMsg?"请输入用户名和密码":requestScope.errorMsg}>
            </div>
        </div>
        <div class="form">
            <form action="userService" method="post">
                <input type="hidden" name="action" value="login">
                <label>用户名称: </label>
                <input class="itxt" type="text" placeholder="请输入用户名" autocomplete="off" tabindex="1"
                    name="username"/>
                <br/>
                <br/>
                <label>用户密码: </label>
            </form>
        </div>
    </div>
</div>
</body>
</html>
```

```

        <input class="itxt" type="password" placeholder="请输入
密码" autocomplete="off" tabindex="1"
        name="password" />
        <br/>
        <br/>
        <input type="submit" value="登录" id="sub_btn" />
    </form>
</div>

</div>
</div>
</div>
<!-- 静态包含页脚信息 -->
<%@ include file="/pages/common/footer.jsp"%>
</body>
</html>

```

注册

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>会员注册页面</title>
    <!-- 静态包含 base css样式 JQuery -->
    <%@ include file="/pages/common/header.jsp"%>
    <script type="text/javascript">
        // 页面加载完成之后
        $(function () {
            // 给注册绑定单击事件
            $("#sub_btn").click(function () {
                // 验证用户名：必须由字母，数字下划线组成，并且长度为5到12位
                //1 获取用户名输入框里的内容
                var usernameText = $("#username").val();
                //2 创建正则表达式对象
                var usernamePatt = /^\\w{5,12}$/;
                //3 使用test方法验证
                if (!usernamePatt.test(usernameText)) {
                    //4 提示用户结果
                    $("span.errorMsg").text("用户名不合法!");

                    return false;
                }
            })
        })
    </script>

```



```

// 验证密码：必须由字母，数字下划线组成，并且长度为5到12位
//1 获取用户名输入框里的内容
var passwordText = $("#password").val();
//2 创建正则表达式对象
var passwordPatt = /^w{5,12}$/;
//3 使用test方法验证
if (!passwordPatt.test(passwordText)) {
    //4 提示用户结果
    $("span.errorMsg").text("密码不合法!");

    return false;
}

// 验证确认密码：和密码相同
//1 获取确认密码内容
var repwdText = $("#repwd").val();
//2 和密码相比较
if (repwdText != passwordText) {
    //3 提示用户
    $("span.errorMsg").text("确认密码和密码不一致!");

    return false;
}

// 邮箱验证：xxxxxx@xxx.com
//1 获取邮箱里的内容
var emailText = $("#email").val();
//2 创建正则表达式对象
var emailPatt = /^[a-z\d]+(\.[a-z\d]+)*@([\da-z](-[\da-z])?)+
(\.{1,2}[a-z]+)$/;
//3 使用test方法验证是否合法
if (!emailPatt.test(emailText)) {
    //4 提示用户
    $("span.errorMsg").text("邮箱格式不合法!");

    return false;
}

// 验证码：现在只需要验证用户已输入。因为还没讲到服务器。验证码生成。
var codeText = $("#code").val();

//去掉验证码前后空格
// alert("去空格前: [" +codeText+"]")
codeText = $.trim(codeText);
// alert("去空格后: [" +codeText+"]")

if (codeText == null || codeText == "") {
    //4 提示用户
    $("span.errorMsg").text("验证码不能为空!");
}

```

```

        return false;
    }

    // 去掉错误信息
    $("span.errorMsg").text("");

});

});

</script>
<style type="text/css">
    .login_form {
        height: 420px;
        margin-top: 25px;
    }

</style>
</head>
<body>
<div id="login_header">
    
</div>

<div class="login_banner">

    <div id="l_content">
        <span class="login_word">欢迎注册</span>
    </div>

    <div id="content">
        <div class="login_form">
            <div class="login_box">
                <div class="tit">
                    <h1>注册会员</h1>
                    <span class="errorMsg">
<%--
                        <%=
request.getAttribute("errorMsg")==null?"":request.getAttribute("errorMsg")%>--
%>

                        ${requestScope.errorMsg}
                    </span>
                </div>
            </div>
            <div class="form">
                <form action="userServlet" method="post">
                    <input type="hidden" name="action" value="regist">
                    <label>用户名称: </label>
                    <input class="itxt" type="text" placeholder="请输入用户
名"

```

```

        autocomplete="off" tabindex="1" name="username"
id="username"

        value="{requestScope.username}"/>
<br/>
<br/>
<label>用户密码: </label>
<input class="itxt" type="password" placeholder="请输入
密码"

        autocomplete="off" tabindex="1" name="password"
id="password"/>

<br/>
<br/>
<label>确认密码: </label>
<input class="itxt" type="password" placeholder="确认密
码"

        autocomplete="off" tabindex="1" name="repwd"
id="repwd"/>

<br/>
<br/>
<label>电子邮件: </label>
<input class="itxt" type="text" placeholder="请输入邮箱
地址"

        autocomplete="off" tabindex="1" name="email"
id="email"

        value="{requestScope.email}"/>
<br/>
<br/>
<label>验证码: </label>
<input class="itxt" type="text" name="code"
style="width: 150px;" id="code"/>

<br/>
<br/>
<input type="submit" value="注册" id="sub_btn"/>
</form>
</div>

</div>
</div>
</div>
<!-- 静态包含页脚信息 --%>
<%@ include file="/pages/common/footer.jsp"%>
</body>
</html>

```

