浙江工业大学

Web 应用开发上机作业

2022 级

实验题目： 课后作业 3

学生姓名：李飞飞

学生学号：2022

任课教师：赵 小 敏

专业班级：2022 软件工程（移动应用开发 1）

学 院：计算机科学与技术学院

2024 年 3 月 29 日

目　　录

1、水费计算

1.1 题目要求：

写一个 input.jsp 为某市民输入一年六次抄水表的用水量，提交后有 waterfee.jsp 计算该市民一年每次需缴纳的水费和累计一年的水费，并将结果在页面输出。要求判断输入的六次用水量的有效性，要求都为整数，否则提示输入的数据无效

1.2 需求分析

某市民希望能够输入一年内六次的抄水表用水量，然后系统能够计算出该市民一年内每次需要缴纳的水费以及累计一年的水费，并将结果在页面上输出。

1.2.1 功能描述

- **输入页面（input.jsp）**
  - 提供六个输入框，用于输入六次抄水表的用水量。
  - 提供提交按钮用于提交表单数据。
  - 如果输入的数据无效，提示用户输入的数据无效。
- **水费计算页面（waterfee.jsp）**
  - 接收输入页面提交的数据。
  - 判断输入的用水量是否有效，必须为整数。
  - 计算并输出一年内累计的水费。

1.2.2 业务处理流程及要求

- 输入页面（input.jsp）：

用户访问输入页面。

用户输入六次抄水表的用水量。

用户点击提交按钮，若数据不合法报错。

- 水费计算页面（waterfee.jsp）：

接收输入页面提交的数据。

对用户输入的每次用水量进行有效性验证：

计算并输出一年内累计的水费。

1.2.3 输入信息

六次水表用水量

1.2.4 输出信息

总价或者报错

1.3 开发步骤

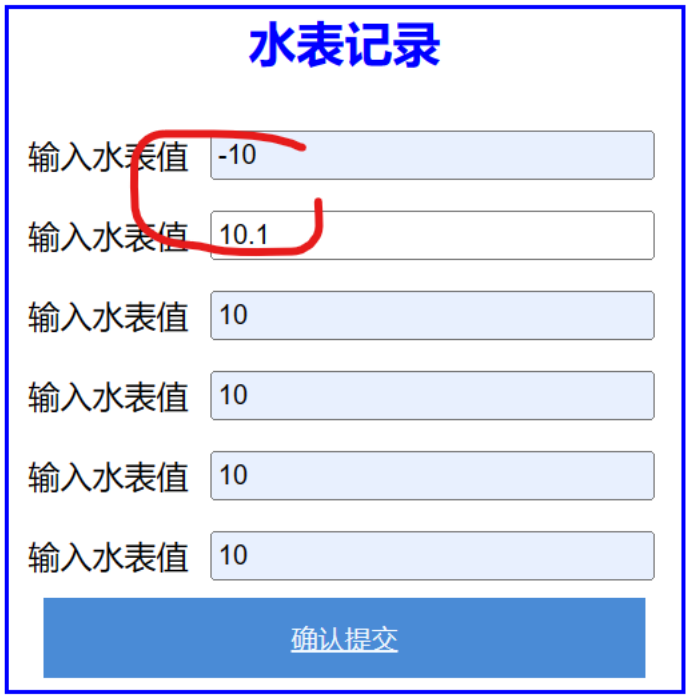
编写 input.jsp->编写 waterfree.jsp->

1.4 运行结果

- 输入页面

- 数据报错


localhost:8080 显示
请输入正确的数值!
确定


水表记录
输入水表值 -10
输入水表值 10.1
输入水表值 10
输入水表值 10
输入水表值 10
输入水表值 10
确认提交

- 计算结果——阶梯 1

- 计算结果——阶梯 2

- 计算结果——阶梯 3

1.5 问题及解决方案

无

1.6 实验收获

无

1.7 源代码

- Input.jsp

```
<%--
  Created by IntelliJ IDEA.
  User: Li Feifei
  Date: 2024/3/27
  Time: 4:22
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>登录页面</title>
    <style type="text/css">
        .myTable{
            border: 2px solid #0000FF;
            margin-top: 15%;
        }
        .inputs{
```

```
            width: 95%;
            height: 25px;
            color: #999;
        }
        .btn{
            background-color:#4a8bd6;
            width: 90%;
            height: 40px;
            border: none;
            color: aliceblue;
            text-decoration: underline; /* 添加白色下划线 */
            margin: 10px;
        }
    </style>
</head>
<body>
<form onSubmit="return custCheck()" action="../firstServlet.do"
method="post">
    <table rules=none cellspacing=0  width=340 align=center
class="myTable">
        <!-- 提示 -->
        <tr height="40">
            <td colspan="2">
                <h2 align="center">
                    <font color="blue">用户登录</font>
                </h2>
            </td>
        </tr>
        <!-- 用户名 -->
        <tr>
            <td height=40 align="center" width=80>用   
户   名</td>
            <td>
                <input class="inputs"
                       placeholder="请输入用户名"

onFocus="if(value==defaultValue){value='';this.style.color='#000'}"
                       onBlur="if(!value){value=defaultValue;
this.style.color='#999'}"
                       name="userName" id="userName">
            </td>
        </tr>
        <!-- 密码 -->
        <tr>
```

```
            <td height=40 align="center" width=80>密
          码</td>
            <td height=40>
                <input class="inputs"
                       placeholder="请输入密码"

onFocus="if(value==defaultValue){value='';this.style.color='#000'}"
                       onBlur="if(!value){value=defaultValue;
this.style.color='#999'}"
                       name="password" id="password">
            </td>
        </tr>

        <!-- 登录 -->
        <tr height="40">
            <td colspan="2" align="center">
                <button width="340" height="40" class="btn" >立即登录
</button>
            </td>
        </tr>

    </table>
</form>
<script language="JavaScript" type="text/javascript">
    function custCheck() {
        var userName = document.getElementById("userName");
        var password = document.getElementById("password");
        if (userName.value === "" || userName.value === "请输入用户名
") {
            alert("用户名不能为空!");
            return false;
        } else if (password.value === "" || password.value === "请输入
密码") {
            alert("密码不能为空!");
            return false;
        }
        return true;
    }
</script>

</body>
</html>
```

- Waterfree.jsp

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>WaterFree</title>
</head>
<body>
<h1>水费计算结果</h1>
    <%
        // 获取水表记录页面提交的值
        int value1 = Integer.parseInt(request.getParameter("value1"));
        int value2 = Integer.parseInt(request.getParameter("value2"));
        int value3 = Integer.parseInt(request.getParameter("value3"));
        int value4 = Integer.parseInt(request.getParameter("value4"));
        int value5 = Integer.parseInt(request.getParameter("value5"));
        int value6 = Integer.parseInt(request.getParameter("value6"));

        // 计算总用水量
        int totalWater = value1 + value2 + value3 + value4 + value5 +
value6;

        // 假设水费计算规则为每立方米水费用为 5 元
        double unitPrice;
        double disposalPrice=1.0;
        if(totalWater<=216) unitPrice=1.9;
        else if(totalWater>300) unitPrice=5.7;
        else  unitPrice=2.85;
        double totalFee=(unitPrice+disposalPrice)*totalWater;
    %>

    <p>总用水量: <%= totalWater %> 吨</p>
    <p>水费单价: <%= unitPrice+disposalPrice %> 元/吨</p>
    <p>总水费: <%= totalFee %> 元</p>
</body>
</html>
```

2.用户登录页面

2.1 题目要求

假设用户登录页面 login.jsp 在 web 工程 web2024 的 hw 目录下，即访问 login.jsp 的 URL 地址是 http://localhost:8080/web2024/hw/login.jsp，将 login.jsp 提交的用户名和密码由 FirstServlet（映射地址为/servlet/firstServlet.do）获取进行判断，如果登录成功，则跳转至 SecondServlet（映射地址为/servlet/secondServlet.do）显示用户名信息，如果登录不成功，则跳转至登录失败页面 hw/failed.jsp。其中，用户名和密码分别为 admin、A1356@8ag#

2.2 需求分析

用户希望能够通过输入正确的用户名和密码登录系统，并且能够在登录成功后看到个人信息。提供登录页面并分别处理登录成功与失败情况，成功展示用户信息，失败则返回错误页面

3.2.1 功能描述

- 用户登录页面 (login.jsp)

  提供用户名和密码输入框。

  提供登录按钮用于提交表单数据。

- 登录验证逻辑

  用户提交表单后，用户名和密码将由 FirstServlet 处理。

  如果用户名和密码匹配成功，则重定向至 SecondServlet，显示用户信息。

  如果用户名和密码匹配失败，则重定向至失败页面 failed.jsp。

- SecondServlet

  显示登录成功后的用户信息。

- 失败页面 (failed.jsp)

  显示登录失败信息

3.2.2 业务处理流程及要求

用户输入账号密码->点击确认登录->服务器返回结果

用户访问登录页面，输入用户名和密码。

用户点击登录按钮，表单数据被提交至 FirstServlet 进行验证。

如果验证成功，重定向至 SecondServlet 显示用户信息。

如果验证失败，重定向至失败页面显示失败信息。

3.2.3 输入信息

用户账号密码

3.2.4 输出信息

是否登录成功已经登录成功的用户信息

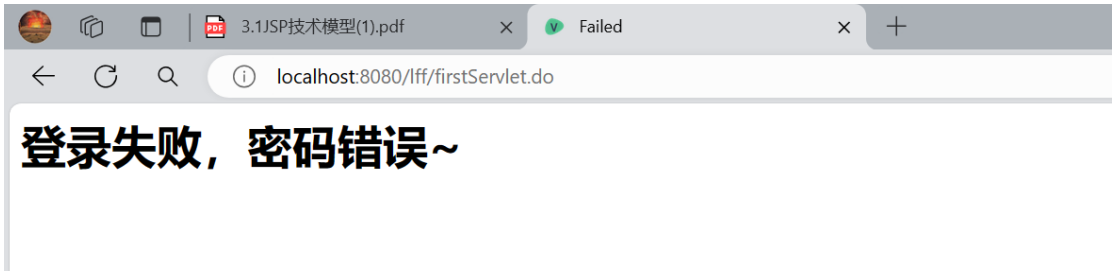
2.3 开发步骤

编写 login.jsp->编写 firstServlet->测试是否正常跳转->编写 secondServlet 以及 failed.jsp 展示登录结果

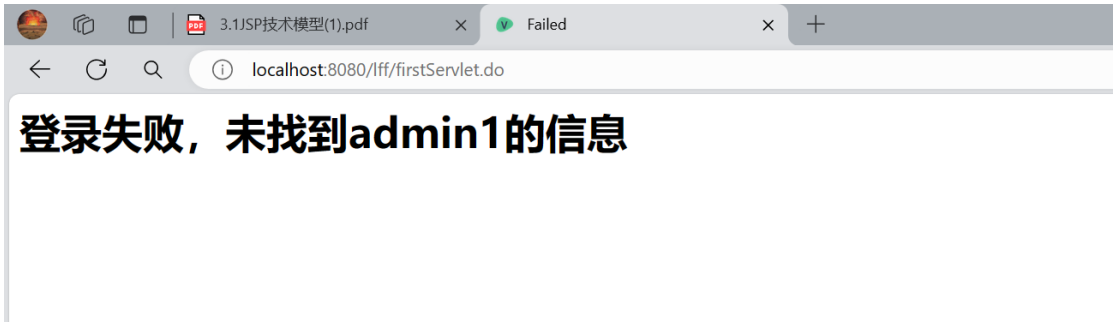
2.4 运行结果

- 登录页面

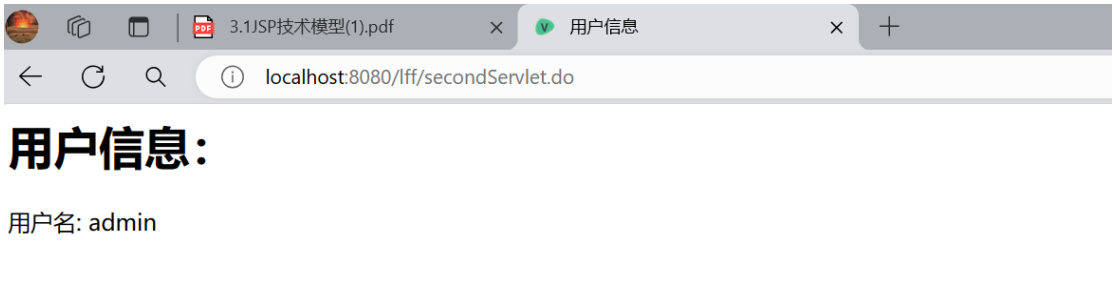
- 登录失败——密码错误



- 登录失败——用户不存在



- 登录成功



2.5 问题及解决方案

从 longin.jsp 跳转 firstServlet 时 404，改成相对路径后解决(action="../firstServlet.do")

2.6 实验收获

1、练习了 servlet 与 jsp 请求转发

2.7 源代码

- firstServlet.do

```
package zjut.edu.homework3;
/*
 * @ Author      : Li Feifei
 * @ Date        : Created in 6:34 2024/3/29
 * @ Description: ${description}
 */

import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

import java.io.IOException;

@WebServlet(name = "firstServlet", value = "/firstServlet.do")
public class firstServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {

    }

    @Override
```

```
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        String userName=request.getParameter("userName");
        String password=request.getParameter("password");
        if(userName.equals("admin")){
            if(password.equals("A1356@8ag#")){
                HttpSession session = request.getSession();
                session.setAttribute("userName", userName);
                // 重定向到 secondServlet
                response.sendRedirect("secondServlet.do");
            }
            else {
                request.setAttribute("errorMsg","登录失败,密码错误~");
                RequestDispatcher rd =
request.getRequestDispatcher("hw/failed.jsp");
                rd.forward(request, response);
            }
        }
        else{
            request.setAttribute("errorMsg","登录失败, 未找到
"+userName+"的信息");
            RequestDispatcher rd =
request.getRequestDispatcher("hw/failed.jsp");
            rd.forward(request, response);
        }
    }
}
```

- secondServlet.do

```
package zjut.edu.homework3;
/*
 * @ Author     : Li Feifei
 * @ Date       : Created in 6:34 2024/3/29
 * @ Description: ${description}
 */

import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

import java.io.IOException;
import java.io.PrintWriter;
```

```java
@WebServlet(name = "secondServlet", value = "/secondServlet.do")
public class secondServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        String userName = (String) session.getAttribute("userName");
        if(userName != null) {
            // 如果用户信息存在，展示用户信息
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            out.println("<html>");
            out.println("<head><title>用户信息</title></head>");
            out.println("<body>");
            out.println("<h1>用户信息：</h1>");
            out.println("<p>用户名：" + userName + "</p>");
            out.println("</body>");
            out.println("</html>");
        }
    }

    @Override
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {

    }
}
```

- failed.jsp

```jsp
<%--
  Created by IntelliJ IDEA.
  User: Li Feifei
  Date: 2024/3/27
  Time: 4:23
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Failed</title>
</head>
<body>
    <h1><%= request.getAttribute("errorMsg") %></h1>
```

```
</body>
</html>
```

- login.jsp

```
<%--
  Created by IntelliJ IDEA.
  User: Li Feifei
  Date: 2024/3/27
  Time: 4:22
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>登录页面</title>
    <style type="text/css">
        .myTable{
            border: 2px solid #0000FF;
            margin-top: 15%;
        }
        .inputs{
            width: 95%;
            height: 25px;
            color: #999;
        }
        .btn{
            background-color:#4a8bd6;
            width: 90%;
            height: 40px;
            border: none;
            color: aliceblue;
            text-decoration: underline; /* 添加白色下划线 */
            margin: 10px;
        }
    </style>
</head>
<body>
<form onSubmit="return custCheck()" action="../firstServlet.do"
method="post">
    <table rules=none cellspacing=0  width=340 align=center
class="myTable">
```

```
        <!-- 提示 -->
        <tr height="40">
            <td colspan="2">
                <h2 align="center">
                    <font color="blue">用户登录</font>
                </h2>
            </td>
        </tr>
        <!-- 用户名 -->
        <tr>
            <td height=40 align="center" width=80>用   
户   名</td>
            <td>
                <input class="inputs"
                       placeholder="请输入用户名"

onFocus="if(value==defaultValue){value='';this.style.color='#000'}"
                       onBlur="if(!value){value=defaultValue;
this.style.color='#999'}"
                       name="userName" id="userName">
            </td>
        </tr>
        <!-- 密码 -->
        <tr>
            <td height=40 align="center" width=80>密
         码</td>
            <td height=40>
                <input class="inputs"
                       placeholder="请输入密码"

onFocus="if(value==defaultValue){value='';this.style.color='#000'}"
                       onBlur="if(!value){value=defaultValue;
this.style.color='#999'}"
                       name="password" id="password">
            </td>
        </tr>

        <!-- 登录 -->
        <tr height="40">
            <td colspan="2" align="center">
                <button width="340" height="40" class="btn" >立即登录
</button>
            </td>
        </tr>
```

```
    </table>
</form>
<script language="JavaScript" type="text/javascript">
    function custCheck() {
        var userName = document.getElementById("userName");
        var password = document.getElementById("password");
        if (userName.value === "" || userName.value === "请输入用户名
") {
            alert("用户名不能为空!");
            return false;
        } else if (password.value === "" || password.value === "请输入
密码") {
            alert("密码不能为空!");
            return false;
        }
        return true;
    }
</script>

</body>
</html>
```

3.用户登录页面 2

3.1 题目要求

假设用户登录页面 login.jsp 在 web 工程 web2024 的 hw 目录下，即访问 login.jsp 的 URL 地址是 http://localhost:8080/web2024/hw/login.jsp，将 login.jsp 提交的用户名和密码由 FirstServlet（映射地址为/servlet/firstServlet.do）获取进行判断，如果登录成功，则跳转至 SecondServlet（映射地址为/servlet/secondServlet.do）显示用户名信息，如果登录不成功，则跳转至登录失败页面 hw/failed.jsp。要求：用户名和密码存在 WEB-INF/userinfo.txt 文件中，要求密码用国密算法 sm3 加密，若页面出现 404 错误则统一跳转至 404.html 页面，500 错误则统一跳转至 500.html 页面，并提示出错信息。

3.2 需求分析

用户希望能够通过输入正确的用户名和密码登录系统，并根据登录结果进行相应的页面跳转和信息展示。同时，用户希望在出现 404 和 500 错误时，能够友好地提示出错信息并跳转至相应的错误页面。

2.2.1 功能描述

- **用户登录页面 (login.jsp)**
  - 提供用户名和密码输入框。
  - 提供登录按钮用于提交表单数据。
- **登录验证逻辑**
  - 用户提交表单后，用户名和密码将由 FirstServlet 处理。
  - FirstServlet 将从 WEB-INF/userinfo.txt 文件中读取用户名和密码进行验证。
  - 如果登录成功，则重定向至 SecondServlet 显示用户名信息。
  - 如果登录不成功，则重定向至登录失败页面 failed.jsp。
- **密码加密要求**
  - 用户密码需要用国密算法 SM3 加密后存储在 userinfo.txt 文件中。
- **404 和 500 错误处理**
  - 在出现 404 错误时，统一跳转至 404.html 页面，并提示出错信息。
  - 在出现 500 错误时，统一跳转至 500.html 页面，并提示出错信息。

2.2.2 业务处理流程及要求

1. 用户登录流程：

    1. 用户访问登录页面：localhost:8080/lff/hw/login.jsp
    2. 用户输入用户名和密码，点击登录按钮提交表单数据。
    3. 提交的表单数据由 FirstServlet 处理。

2. FirstServlet 处理流程：

    1. FirstServlet 接收到表单数据后，从 WEB-INF/userinfo.txt 文件中读取用户名和加密后的密码信息。
    2. 对比用户提交的用户名和密码是否匹配：
        - 如果匹配成功，重定向至 SecondServlet 显示用户名信息。
        - 如果匹配失败，重定向至登录失败页面 failed.jsp。

3. SecondServlet 处理流程：

    1. SecondServlet 接收到请求后，显示用户名信息。

4. 密码加密要求：

    - 用户密码需要在存储时经过国密算法 SM3 加密处理，与存储在 userinfo.txt 文件中的密码进行匹配。

5. 错误处理要求：

    - 出现 404 错误时，统一跳转至 404.html 页面，并提示页面不存在的信息。
    - 出现 500 错误时，统一跳转至 500.html 页面，并提示服务器内部错误的信息。

2.2.3 输入信息

账号密码信息

2.2.4 输出信息

- 登录成功：

跳转至 SecondServlet，显示用户名信息。

- 登录失败：

跳转至登录失败页面 failed.jsp，提示登录失败信息。

- 404 错误：

跳转至 404.html 页面，提示页面不存在的信息。

- 500 错误：

跳转至 500.html 页面，提示服务器内部错误的信息。

3.3 开发步骤

在实验 2 基础上添加 User 与 UserUtil,分别为封装的用户类，用户工具类。工具类提供三个方法，具体功能见注释。改写 firstServlet

```
import java.util.Objects;
import java.util.Set;

/**
 * @ Author      : Li Feifei
 * @ Date        : Created in 1:42 2024/3/15
 * @ Description: 用户工具类
 */
6 个用法  新 *
public class UsersUtil {

    //获取所有userinfo已被注册用户
    1 个用法  新 *
    public static Set<User> getAllUsers(){...}

    //记录一个用户(已经保证合法)
    0 个用法  新 *
    public static Boolean saveUser(User user){...}

    //获取加密后字符串
    2 个用法  新 *
    public static String passwordEncode(String password){
        MessageDigest digest = new SM3.Digest();
        byte[] hashBytes = digest.digest(password.getBytes());
        return Hex.toHexString(hashBytes).toUpperCase();
    }
}
```

3.4 运行结果

- 用户数据

三个新用户密码分别为 1 2 3

```
admin|ACC5ED6C1BD22CBD9706C1AEAC7536E0ABC857640DBE63376BC0D3D165738E23
lff|CBDDDB8E8421B23498480570D7D75330538A6882F5DFDC3B64115C647F3328C4
gs|A0DC2D74B9B0E3C87E076003DBFE472A424CB3032463CB339E351460765A822E
lch|55E3192D096E62D4F9CD00E734A949DE2B8E55B13D9B85B1D2D2999C9DB2E72C
```

- 用户登录结果

用户信息:

用户名: lff

- 404

- 500

3.5 问题及解决方案

加密算法导入依赖实现

```
<!--        加密-->
<dependency>
    <groupId>org.bouncycastle</groupId>
    <artifactId>bcprov-jdk15on</artifactId>
    <version>1.68</version>
</dependency>
```

3.6 实验收获

学会了通过配置统一处理错误页面

3.7 源代码

与实验 2 重复的部分不放

- User 用了 Lombok

```
package zjut.edu.homework3.Utils;

import lombok.AllArgsConstructor;
```

```java
import lombok.Data;
import lombok.NoArgsConstructor;

/*
 * @ Author     : Li Feifei
 * @ Date       : Created in 6:46 2024/3/29
 * @ Description: 用户类
 */
@Data
@AllArgsConstructor
@NoArgsConstructor
public class User {
    private String userName;
    private String password;
}
```

- UserUtil

```java
package zjut.edu.homework3.Utils;

import org.bouncycastle.jcajce.provider.digest.SM3;
import org.bouncycastle.util.encoders.Hex;

import java.io.*;
import java.security.MessageDigest;
import java.util.HashSet;
import java.util.Objects;
import java.util.Set;

/**
 * @ Author     : Li Feifei
 * @ Date       : Created in 1:42 2024/3/15
 * @ Description: 用户工具类
 */
public class UsersUtil {

    //获取所有 userinfo 已被注册用户
    public static Set<User> getAllUsers(){
        Set<User> users = new HashSet<>();

        //尝试读取 userinfo.txt 文件,不存在创建
        File file = new File("userinfo.txt");
        if (!file.exists()) {
            try {
                file.createNewFile(); // 如果文件不存在，则创建文件
```

```
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

        try (
             BufferedReader reader = new BufferedReader(new
InputStreamReader(

Objects.requireNonNull(UsersUtil.class.getResourceAsStream("/userinfo
.txt"))
        ))
        ) {
            String line;
            while ((line = reader.readLine()) != null) {
                // 按 "|" 分隔行数据
                String[] fields = line.split("\\|");
                if (fields.length == 2)//确保每行数据都包含了所有字段
                    users.add(
                            new User(fields[0], fields[1])
                    );
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        System.out.println(users);
        return users;
    }

    //记录一个用户(已经保证合法)
    public static Boolean saveUser(User user){
        try (BufferedWriter writer =
                     new BufferedWriter(
                             new
FileWriter(UsersUtil.class.getClassLoader().getResource("userinfo.txt
").getPath(),
                                     true))) {
            // 使用 | 分隔字段，并加密密码
            String encryptedPassword =
passwordEncode(user.getPassword());
            String userInfo = user.getUserName() + "|" +
encryptedPassword;

            // 将用户信息写入文件
```

```
            writer.write(userInfo);
            System.out.println("文件位置
"+UsersUtil.class.getClassLoader().getResource("userinfo.txt").getPat
h());
            System.out.println(userInfo);
            writer.newLine(); // 换行
            writer.flush();
            writer.close();
            return true;
        } catch (IOException e) {
            e.printStackTrace();
            return false;
        }
    }

    //获取加密后字符串
    public static String passwordEncode(String password){
        MessageDigest digest = new SM3.Digest();
        byte[] hashBytes = digest.digest(password.getBytes());
        return Hex.toHexString(hashBytes).toUpperCase();
    }
}
```

- firstServlet

```
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    String userName=request.getParameter("userName");
    String password=request.getParameter("password");
    Set<User> st= UsersUtil.getAllUsers();
    for(User user:st){
        if(userName.equals(user.getUserName())){

if(UsersUtil.passwordEncode(password).equals(user.getPassword())){
                HttpSession session = request.getSession();
                session.setAttribute("userName", userName);
                // 重定向到 secondServlet
                response.sendRedirect("secondServlet.do");
                return;
            }
            else {
                request.setAttribute("errorMsg","登录失败,密码错误~");
                RequestDispatcher rd =
request.getRequestDispatcher("hw/failed.jsp");
                rd.forward(request, response);
```

```
                return;
            }
        }

    }
    //未找到
    request.setAttribute("errorMsg","登录失败，未找到"+userName+"的信息");
    RequestDispatcher rd = request.getRequestDispatcher("hw/failed.jsp");
    rd.forward(request, response);
}
```