# 实验4 Java Web开发入门

**一、实验目的**

1、熟悉Java Web开发的一般步骤。

2、理解前端与后端的概念。

**二、实验报告要求**

1、实验报告提交word文档，命名为“学号+姓名+实验4”。

2、本次实验共2题。

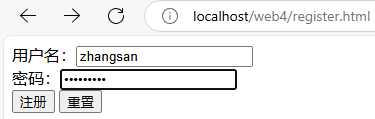
（1）第1题：给出填好**空缺**部分的**完整代码**，以及根据“本人的**姓名全拼**”作为用户名、“本人的**学号**”作为密码进行注册的截图。

注：post请求提交的中文，需要经过处理，才能被服务器正确接收

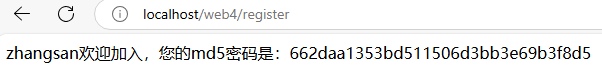
【截图范例】

姓名为“张三”、学号为“226710001”的同学应提供以下截图：

1）注册页面的截图：

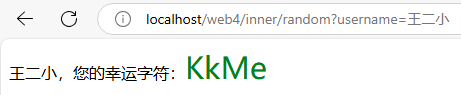


2）注册成功的截图：



（2）第2题：给出填好**空缺**部分的**完整代码**，以及参数为“**本人姓名**”的幸运字符截图。

【截图范例】



3、附上调试程序的过程中，出现的问题、以及解决方法。【可选】

**三、实验内容**

1、设计如图4-1所示的注册页面。

当输入用户名和密码，单击“注册”按钮，如果用户名为“lily”，则在浏览器中输出“用户名已存在，注册失败”，如图4-2所示；

否则，输出 “\*\*\*欢迎加入，您的md5密码是：\*\*\*\*\*\*”，如图4-3所示。

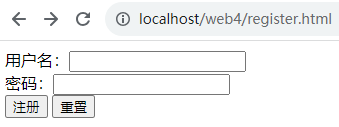


图4-1 注册页面register.html

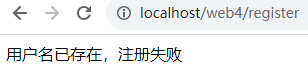


图4-2 注册失败

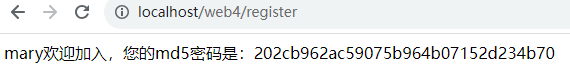


图4-3 注册成功

【实验步骤】

（1）新建一个名为”web4”的项目（Web应用程序）。在项目的“web”节点下，新建HTML，命名为“register”。register.html提供一个注册页面。

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>注册</**title**>  
</**head**>  
<**body**>  
 **<!-- 【代码1】-->**</**body**>  
</**html**>

（2）在项目中新建名为“it.servlet”的Java包，在包中新建名为“RegisterServlet”的Java类。RegisterServlet.java进行注册的验证。如果用户名为“lily”，则在浏览器中输出“用户名已存在，注册失败”；否则输出“\*\*\*欢迎加入。您的md5密码是：\*\*\*\*\*\*”。

**package** it.servlet;  
  
**import** javax.servlet.ServletException;  
**import** javax.servlet.annotation.WebServlet;  
**import** javax.servlet.http.HttpServlet;  
**import** javax.servlet.http.HttpServletRequest;  
**import** javax.servlet.http.HttpServletResponse;  
**import** java.io.IOException;  
**import** java.math.BigInteger;  
**import** java.security.MessageDigest;  
**import** java.security.NoSuchAlgorithmException;  
  
**//【代码2】**

**public class** RegisterServlet **extends** HttpServlet {  
 //md5加密方法  
 **public static** String stringToMD5(String plainText) {  
 **byte**[] secretBytes = **null**;  
 **try** {  
 secretBytes = MessageDigest.getInstance(**"md5"**).  
 digest(plainText.getBytes());  
 } **catch** (NoSuchAlgorithmException e) {  
 **throw new** RuntimeException(**"没有这个md5算法！"**);  
 }

String md5code = **new** BigInteger(1, secretBytes).toString(16);  
 **for** (**int** i = 0; i < 32 - md5code.length(); i++) {  
 md5code = **"0"** + md5code;  
 }  
 **return** md5code;  
 }  
  
 @Override  
 **protected void** doPost(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  
 **//【代码3】**

}  
}

（3）设置项目的Application Context为“/web4”。启动tomcat，部署项目，查看结果。

2、当点击如图4-4所示的页面上的“lucky”图片，将得到显示用户名、以及4个随机字符的页面，如图4-5所示。

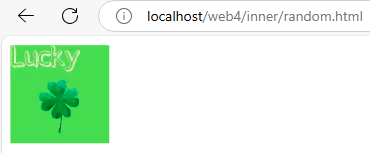


图4-4 超链接图片页面

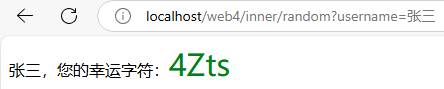
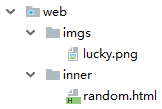


图4-5 返回4个随机字符

【实验步骤】

（1）在“web4”的项目的web节点下，新建2个文件夹，分别名为“inner”和“imgs”。在“inner”文件夹中，创建“random.html”文件；“imgs”文件夹中，存储“lucky.png”图片。



（2）编写random.html页面代码，该页面显示超链接图片lucky.png。

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>随机字符串</**title**>  
</**head**>  
<**body**>  
 **<!--"张三"为虚拟的人名，要求将“张三”替换为本人的姓名-->**<**a href="random?username=张三"**>  
 **<!-- 【代码1】显示lucky.png图片，图片大小设置为100px×100px-->**</**a**>  
</**body**>  
**</html>**

（3）在“web4”项目的“it.servlet”包中新建“RandomServlet”类。该类接收的超链接图片发出的请求，返回一个由大写字母、小写字母以及数字组成的长度为4的随机字符串。

**package** it.servlet;  
  
**import** javax.servlet.ServletException;  
**import** javax.servlet.annotation.WebServlet;  
**import** javax.servlet.http.HttpServlet;  
**import** javax.servlet.http.HttpServletRequest;  
**import** javax.servlet.http.HttpServletResponse;  
**import** java.io.IOException;  
**import** java.util.Random;  
  
**//【代码2】****public class** RandomServlet **extends** HttpServlet {  
 @Override  
 **protected void**  **/\*【代码3】\*/** (HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  
 String str = **"ABCDEFGHIGKLMNOPQRSTUVWXYZabcdefghigklmnopqrstuvwxyz0123456789"**;  
 Random ran = **new** Random();  
 **//【代码4】创建字符串对象s****for** (**int** i = 1; i <= 4; i++) {  
 **//【代码5】从str中随机取得一个字符ch**  
 s.append(ch);   
 }  
 String username = **/\*【代码6】获取参数username的值\*/**;  
 resp.setContentType(**"text/html;charset=utf-8"**);  
 resp.getWriter().println(username + **"，您的幸运字符："** +  
 **"<span style='color: green;font-size: 30px'>"**+  
 s.toString() + **"</span>"**);  
 }  
}