



视频讲解

建议学时：2

表单是用户和服务器之间进行信息交互的重要手段,有了表单,JSP 程序才可以更加丰富多彩。本章将学习 JSP 编程中的表单开发,首先对表单的基本结构和基本属性进行学习,然后学习各种表单元素与服务器的交互,最后对隐藏表单的作用进行讲解。

5.1 认识表单

5.1.1 表单的作用

在编写 JSP 表单之前首先了解一下表单的作用。

以百度为例,若在百度上输入一个关键词,例如“玫瑰花”,如图 5-1 所示。



图 5-1 百度搜索界面

单击“百度一下”按钮,百度能够将所有与“玫瑰花”有关的搜索结果展现出来,很明显,百度在服务器端进行了一个搜索工作。

此处百度提供的输入界面就是一个表单。用户可以在表单上进行一些输入,在提交时可以根据用户的输入来执行相应的程序。

同样,在某系统中如果用户要进行登录,则必须输入账号和密码,如图 5-2 所示。这也是一个表单。所以,表单是可以由用户输入并提交给服务器端的一个图形界面。

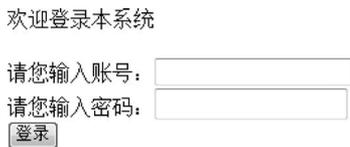


图 5-2 系统登录界面

5.1.2 定义表单

对于表单的定义,在网页制作过程中进行了详细的介绍,在这里仅仅根据 JSP 来介绍表单的基本定义方法。

表单有如下性质。

- (1) 在表单中可以输入一些内容,这些输入功能由控件提供,叫作表单元素。
- (2) 在表单中一般会有一个按钮负责提交。

(3) 单击提交按钮,表单元素中的内容会提交给服务器端。

(4) 表单元素放在< form ></form >之间。

在 MyEclipse 中建立一个项目 Prj05。建立一个页面,5.1.1 节的登录表单可以由 form.jsp 实现,代码如下。

form.jsp

```
<%@ page language = "java" contentType = "text/html; charset = gb2312" %>
<html>
  <body>
    欢迎登录本系统
    <form>
      请您输入账号: <input name = "account" type = "text"><br>
      请您输入密码: <input name = "password" type = "password"><br>
      <input type = "submit" value = "登录">
    </form>
  </body>
</html>
```

运行,得到 5.1.1 节中的登录界面。



问答

问: 表单提交给服务器端,如何确定到底提交给哪一个页面?

答: 可以用< form >中的 action 属性确定。例如:

```
<form action = "page.jsp">
  请您输入账号: <input name = "account" type = "text"><br>
  请您输入密码: <input name = "password" type = "password"><br>
  <input type = "submit" value = "登录">
</form>
```

以上代码表示将表单中输入的内容提交给 page.jsp 去运行。

注意: 此处的 action 值支持相对路径。例如:

- ../page.jsp 表示当前页面的上一级目录中的 page.jsp。
- jsps/page.jsp 表示当前目录 jsps 目录中的 page.jsp。

它也支持绝对路径,例如:

/Prj05/page.jsp 表示 Prj05 中的 page.jsp。

问: page.jsp 如何获取提交过来的值?

答: 方法是用 request 对象。例如:

```
<%
  //获取表单中 name = account 的表单元素中输入的值,赋值给 str
  String str = request.getParameter("account");
%>
```

如果表单中没有 name=account 的表单元素,str 为 null;如果在表单元素 account 中没有输入任何内容就提交,str 为 ""。

问: <input type="submit" value="登录">表示提交按钮,可以用普通按钮吗?

答：不可以，如果将该按钮改为“< input type="button" value="登录">”，虽然显示效果一样，但是单击没有提交功能。当然，可以用 JavaScript 进行提交。

5.2 单一表单元素数据的获取

单一表单元素是指表单元素的值送给服务器端时仅仅是一个变量，这种情况下的表单元素主要有文本框、密码框、多行文本框、单选按钮和下拉菜单等。

5.2.1 获取文本框中的数据

例如，在学生管理系统中用户可以模糊查询学生信息，输入学生姓名的部分资料，就可以显示学生的信息，此时表单中可以包含一个文本框，实现代码如 textForm.jsp 所示。

textForm.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
<html>
  <body>
    <form action = "textForm_result.jsp">
      请您输入学生的模糊资料: <br>
      <input name = "stuname" type = "text">
      <input type = "submit" value = "查询">
    </form>
  </body>
</html>
```

其运行效果如图 5-3 所示。

“< form action="textForm_result.jsp">”说明将页面提交到 textForm_result.jsp, textForm_result.jsp 的代码如下。

请您输入学生的模糊资料:

图 5-3 模糊查询界面

textForm_result.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
<html>
  <body>
    <%
      String stuname = request.getParameter("stuname");
      out.println("输入的查询关键字为:" + stuname);
    %>
  </body>
</html>
```

输入的查询关键字为:Rose

图 5-4 模糊查询结果界面

输入一个关键字，例如“Rose”，单击“查询”按钮，能够运行 textForm_result.jsp, 效果如图 5-4 所示。

在实际项目中应该根据这个关键字查询数据库，此处省略。

特别提醒：

(1) 如果输入的是“罗斯”，提交后页面显示如图 5-5 所示。

这表示中文无法显示，对于该问题的解决，在本章最后会作讲解。

(2) 输入“Rose”之后提交,浏览器的地址栏上出现的效果如图 5-6 所示。

输入的查询关键字为:????

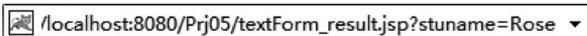


图 5-5 结果界面

图 5-6 浏览器显示界面

这说明提交的内容能够在浏览器的地址栏上看到。很显然不安全,怎样解决?方法是在表单中将 method 属性设置为 post,也就是将 textForm.jsp 中的表单改为如下格式。

textForm.jsp

```
...
< form action = "textForm_result.jsp" method = "post">
    请您输入学生的模糊资料: <br >
    < input name = "stuname" type = "text">
    < input type = "submit" value = "查询">
</form >
...
```

注意:在默认情况下是 get 方式,get 方式和 post 方式是提交请求的两种常见方式。

5.2.2 获取密码框中的数据

在很多界面中都用到了密码。比如用户注册时需要输入自己的密码,然后提交,最后被系统添加到数据库中。下面的 passwordForm.jsp 实现这个功能,代码如下。

passwordForm.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
< html >
  < body >
    请您输入自己的信息进行注册
    < form action = "passwordForm_result.jsp" method = "post">
      请您输入账号: < input name = "account" type = "text"><br >
      请您输入密码: < input name = "password" type = "password"><br >
      < input type = "submit" value = "注册">
    </form >
  </body >
</html >
```

其运行效果如图 5-7 所示。

请您输入自己的信息进行注册

请您输入账号:

请您输入密码:

图 5-7 注册界面

在实际项目中还应该输入一个确认密码,此处省略。

“< form action = "passwordForm_result.jsp" method = "post">”说明将页面提交到 passwordForm_result.jsp,passwordForm_result.jsp 的代码如下。

passwordForm_result.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
<html>
  <body>
    <%
      String password = request.getParameter("password");
      out.println("密码为:" + password);
    %>
  </body>
</html>
```

输入一个密码,例如“fdtj;df”,然后单击“查询”按钮,能够运行 passwordForm_result.jsp,效果如图 5-8 所示。

密码为:fdtj;df

图 5-8 运行效果界面

在实际项目中这个密码可能被送到数据库,不会显示出来,这里只是举一个简单的例子。

5.2.3 获取多行文本框中的数据

在注册界面中添加一个多行文本框,让用户输入自己的信息。textareaForm.jsp 用来实现这个功能,代码如下。

textareaForm.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
<html>
  <body>
    请您输入自己的信息进行注册
    <form action = "textareaForm_result.jsp" method = "post">
      请您输入账号: <input name = "account" type = "text"><br>
      请您输入密码: <input name = "password" type = "password"><br>
      请您输入个人信息: <br>
      <textarea name = "info" rows = "5" cols = "30"></textarea>
      <input type = "submit" value = "注册">
    </form>
  </body>
</html>
```

其运行效果如图 5-9 所示。

请您输入自己的信息进行注册

请您输入账号:

请您输入密码:

请您输入个人信息:

I am a student.

图 5-9 包含个人信息的注册界面

其中,“I am a student.”是在运行完毕之后手工输入的。

“< form action = " textareaForm _ result. jsp " method = " post " >”说明将页面提交到 textareaForm _ result. jsp, textareaForm _ result. jsp 的代码如下。

textareaForm_result.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
<html>
  <body>
    <%
      String info = request.getParameter("info");
      out.println("个人信息为:" + info);
    %>
  </body>
</html>
```

单击“注册”按钮能够运行 textareaForm_result.jsp,效果如图 5-10 所示。

个人信息为:I am a student.

图 5-10 运行效果界面

5.2.4 获取单选按钮中的数据

在注册界面中设置两个单选按钮,让用户能够选择自己的性别。radioForm.jsp 用来实现这个功能,代码如下。

radioForm.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
<html>
  <body>
    请您输入自己的信息进行注册
    <form action = "radioForm_result.jsp" method = "post">
      请您输入账号: <input name = "account" type = "text"><br>
      请您输入密码: <input name = "password" type = "password"><br>
      请您选择性别:
      <input name = "sex" type = "radio" value = "boy" checked>男
      <input name = "sex" type = "radio" value = "girl">女<br>
      <input type = "submit" value = "注册">
    </form>
  </body>
</html>
```

其运行效果如图 5-11 所示。

请您输入自己的信息进行注册

请您输入账号:
请您输入密码:
请您选择性别: 男 女

图 5-11 包含性别选择的注册界面

“< form action = " radioForm _ result. jsp " method = " post " >”说明将页面提交到

radioForm_result.jsp,radioForm_result.jsp 的代码如下。

radioForm_result.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
<html>
  <body>
    <%
      String sex = request.getParameter("sex");
      out.println("性别为:" + sex);
    %>
  </body>
</html>
```

选择“女”,单击“注册”按钮,能够运行 radioForm_result.jsp,效果如图 5-12 所示。

性别为:girl

图 5-12 运行效果界面

5.2.5 获取下拉菜单中的数据

在注册界面中设置一个下拉菜单,让用户能够选择自己的家乡,代码如 selectForm.jsp 所示。

selectForm.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
<html>
  <body>
    请您输入自己的信息进行注册
    <form action = "selectForm_result.jsp" method = " post " >
      请您输入账号: <input name = "account" type = "text"><br>
      请您输入密码: <input name = "password" type = "password"><br>
      请您选择家乡:
      <select name = "home">
        <option value = "beijing">北京</option>
        <option value = "shanghai">上海</option>
        <option value = "guangdong">广东</option>
      </select>
      <input type = "submit" value = "注册">
    </form>
  </body>
</html>
```

其运行效果如图 5-13 所示。

请您输入自己的信息进行注册

请您输入账号:
请您输入密码:
请您选择家乡: 北京 ▾

图 5-13 包含家乡选择的注册界面

“<form action = " selectForm_result.jsp " method = " post ">”说明将页面提交到

selectForm_result.jsp,selectForm_result.jsp 的代码如下。

selectForm_result.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
<html >
  <body >
    <%
      String home = request.getParameter("home");
      out.println("家乡为:" + home);
    %>
  </body >
</html >
```

选择“上海”,单击“注册”按钮,能够运行 selectForm_result.jsp,效果如图 5-14 所示。

家乡为:shanghai

图 5-14 运行效果界面

5.3 捆绑表单元素数据的获取

捆绑表单元素是指多个同名表单元素的值送给服务器端时是一个捆绑的数组。这种情况下的表单元素主要有复选框、多选列表框、其他同名表单元素等。

此时可以用如下方法得到捆绑的数组。

```
<%
  //获得表单中 name = pName 的表单元素中输入的值,赋值给 str 数组
  String[] str = request.getParameterValues("pName");
%>
```

5.3.1 获取复选框中的数据

在很多系统中用户可以进行注册,例如在注册过程中有 4 个爱好供用户选择,如图 5-15 所示。

请选择您的爱好: 唱歌 跳舞 打球 打游戏

图 5-15 爱好选择示例

用户可以选择,也可以不选择;可以选择全部,也可以选择一部分。此时可以为这几个复选框取同样的名字,作为捆绑数组传给服务器端。下面的 checkForm.jsp 实现了这个功能,代码如下。

checkForm.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
<html >
  <body >
    请您输入自己的信息进行注册
    <form action = "checkForm_result.jsp" method = "post">
      请选择您的爱好:
```

```
< input name = "fav" type = "checkbox" value = "sing">唱歌  
< input name = "fav" type = "checkbox" value = "dance">跳舞  
< input name = "fav" type = "checkbox" value = "ball">打球  
< input name = "fav" type = "checkbox" value = "game">打游戏< br >  
< input type = "submit" value = "注册">  
</form >  
</body >  
</html >
```

其运行效果如图 5-16 所示。

请您输入自己的信息进行注册

请选择您的爱好: 唱歌 跳舞 打球 打游戏

图 5-16 包含爱好选择的注册界面

其中,“唱歌”“跳舞”和“打游戏”是在运行之后手工选择的。

“< form action = " checkForm _ result. jsp" method = " post ">”说明将页面提交到 checkForm_result.jsp,checkForm_result.jsp 的代码如下。

checkForm_result.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>  
<html >  
<body >  
<%  
String[] fav = request.getParameterValues("fav");  
out.println("爱好为:");  
for(int i = 0;i < fav.length;i++){  
out.println(fav[i]);  
}  
%>  
</body >  
</html >
```

在图 5-16 所示的界面中单击“注册”按钮能够运行 checkForm_result.jsp,效果如图 5-17 所示。

爱好为: sing dance game

图 5-17 运行效果界面

5.3.2 获取多选列表框中的数据

5.3.1 节中的需求功能也可以用多选列表框代替,代码如 listForm.jsp 所示。

listForm.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>  
<html >  
<body >  
请您输入自己的信息进行注册  
<form action = "listForm_result.jsp" method = "post">
```

```

    请选择您的爱好: <br>
    <select name = "fav" multiple>
        <option value = "sing">唱歌</option>
        <option value = "dance">跳舞</option>
        <option value = "ball">打球</option>
        <option value = "game">打游戏</option>
    </select>
    <input type = "submit" value = "注册">
</form>
</body>
</html>

```

请您输入自己的信息进行注册



图 5-18 包含多种爱好选择的注册界面

其运行效果如图 5-18 所示。

其中，“唱歌”、“跳舞”和“打游戏”是在运行之后手工选择的(在选择的同时按下 Ctrl 键可以多选)。

“< form action="listForm_result.jsp" method="post">”说明将页面提交到 listForm_result.jsp, listForm_result.jsp 的代码如下。

listForm_result.jsp

```

<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
<html>
  <body>
    <%
      String[] fav = request.getParameterValues("fav");
      out.println("爱好为:");
      for(int i = 0; i < fav.length; i++){
        out.println(fav[i]);
      }
    %>
  </body>
</html>

```

在图 5-18 所示的界面中单击“注册”按钮,能够运行 listForm_result.jsp,效果如图 5-19 所示。

爱好为: sing dance game

图 5-19 运行效果界面

5.3.3 获取其他同名表单元素中的数据

在很多情况下,其他表单元素也可以设置为同名。例如在注册界面上输入用户的电话号码,最多可以输入 4 个,则可以用 4 个同名的文本框进行输入,代码如 multiNameForm.jsp 所示。

multiNameForm.jsp

```

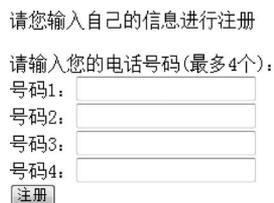
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
<html>
  <body>

```

```
请您输入自己的信息进行注册
< form action = "multiNameForm_result.jsp" method = "post">
    请输入您的电话号码(最多4个): <br>
    <% for(int i = 1;i <= 4;i++){ %>
        号码<% = i %>: <input name = "phone" type = "text"><br>
    <% } %>
    <input type = "submit" value = "注册">
</form>
</body>
</html>
```

注意：此处的4个文本框名字都叫作 phone。

其运行效果如图 5-20 所示。



请您输入自己的信息进行注册

请输入您的电话号码(最多4个):

号码1:

号码2:

号码3:

号码4:

图 5-20 获取多个同名表单的注册界面

其中的号码是需要手工输入的。

“< form action = " multiNameForm_result.jsp " > method = " post " ”说明将页面提交到 multiNameForm_result.jsp, multiNameForm_result.jsp 的代码如下。

multiNameForm_result.jsp

```
<% @ page language = "java" contentType = "text/html; charset = gb2312" %>
< html >
< body >
< %
    String[] phone = request.getParameterValues("phone");
    out.println("号码为:");
    for(int i = 0;i < phone.length;i++){
        out.println(phone[i]);
    }
< % >
</body >
</html >
```

在图 5-20 所示的界面中输入号码,单击“注册”按钮,能够运行 multiNameForm_result.jsp,效果如图 5-21 所示。

号码为: 78954788 75415625 48956425 84587569

图 5-21 运行效果界面

此时,第 1 个号码放在 phone[0]内,第 2 个号码放在 phone[1]内,以此类推。那么到底哪个号码放在哪个位置呢?答案是以文本框在源代码中出现的顺序从数组的头上开始向后放置。

5.4 隐藏表单

前面的章节已经讲过 HTTP 是无状态的协议,在页面之间传递值时必须通过服务器。通过 URL 传值方法可以实现传值。

这里仍以 4.6 节中的例子说明。在页面 1 中定义了一个数值变量,并显示其平方,要求在页面 2 中显示其立方。很明显,页面 2 必须知道页面 1 中定义的那个变量。此时可以用 URL 传值,但是通过 URL 传值方法传递的数据可能被看到。为了避免这个问题,用户可以用表单将页面 1 中的变量传给页面 2。因此,4.6 节中的例子可以通过 formP1.jsp 来实现,代码如下。

formP1.jsp

```
<% @ page language = "java" import = "java.util. * " pageEncoding = "gb2312" %>
<%
    //定义一个变量
    String str = "12";
    int number = Integer.parseInt(str);
%>
该数字的平方为: <%= number * number %><hr>
<form action = "formP2.jsp">
    <input type = "text" name = "number" value = "<%= number %>">
    <input type = "submit" value = "到达 P2">
</form>
```

该数字的平方为: 144

12

图 5-22 formP1.jsp 运行效果

其运行效果如图 5-22 所示。

可以看到,这里实际上是将 number 的值放入表单元素传到下一个页面。但是,number 的值在界面上会被看到,为了既传值又不被看到,可以使用隐藏表单。

在网页制作中 input 有一个“type="hidden"”的选项,它是隐藏在网页中的一个表单元素,并不在网页中显示出来,于是该例的代码可以修改为如 formP1_hidden.jsp 所示的代码。

formP1_hidden.jsp

```
<% @ page language = "java" import = "java.util. * " pageEncoding = "gb2312" %>
<%
    //定义一个变量
    String str = "12";
    int number = Integer.parseInt(str);
%>
该数字的平方为: <%= number * number %><hr>
<form action = "formP2.jsp">
    <input type = "hidden" name = "number" value = "<%= number %>">
    <input type = "submit" value = "到达 P2">
</form>
```

其运行效果如图 5-23 所示。

这样传的值就被隐藏起来了。下面是 formP2.jsp 的代码。

该数字的平方为：144

到达P2

图 5-23 formP1_hidden.jsp 运行效果

formP2.jsp

```
<% @ page language = "java" import = "java.util. * " pageEncoding = "gb2312" % >
<%
    //获得 number
    String str = request.getParameter("number");
    int number = Integer.parseInt(str);
%>
该数字的立方为：<%= number * number * number %><hr >
```

单击 formP1_hidden.jsp 中的按钮,到达 formP2.jsp,效果如图 5-24 所示。

但是,此时浏览器的地址栏上的地址仍带 number 值,如图 5-25 所示。

该数字的立方为：1728



图 5-24 formP2.jsp 运行效果

图 5-25 地址仍带 number 值

此时数据还是能够被看到。

解决该问题的方法是 will form 的 action 属性设置为 post(默认为 get),于是代码变为 formP1_post.jsp 所示的代码。

formP1_post.jsp

```
...
<form action = "formP2.jsp" action = "post">
    <input type = "hidden" name = "number" value = "<%= number %>">
    <input type = "submit" value = "到达 P2">
</form>
...
```

再单击按钮,在 formP2.jsp 中显示结果,但是浏览器的地址栏上的 URL 如图 5-26 所示。



图 5-26 地址栏上的 URL

这说明可以顺利地实现值的传递,并且无法看到传递的信息。

使用该方法有如下问题:

(1) 和 URL 传值方法类似,该方法传输的数据只能是字符串,对数据类型有一定的限制。

(2) 传输数据的值虽然在浏览器的地址栏内不被看到,但是在客户端源代码里面也会被看到。例如,在 formP1.jsp 例子中打开其源代码,如图 5-27 所示。

在“<input type="hidden" name="number" value="12">”中要传递的 number 值被显示出来了,因此从保密的角度讲这也是不安全的,特别是对秘密性要求很严格的数据(例如密码),不推荐用表单方法来传值。

```

1
2
3 该数字的平方为: 144<HR>
4 <form action="formP2.jsp" method="get">
5   <input type="hidden" name="number" value="12">
6   <input type="submit" value="到达p2">
7 </form>

```

图 5-27 客户端源代码

请您输入张海的语文成绩(可修改):

输入成绩:

图 5-28 运行效果界面

同样,表单传值方法也并不是一无是处,由于其简单性和平台支持的多样性,很多程序用表单传值比较方便。下面研究如图 5-28 所示的界面。

在该表单中将成绩输入之后,系统如何知道该成绩是张海的语文成绩呢?换句话说,系统如何知道要修改表中的哪一行呢?如下程序可以将张海的学号(例如 0015)和语文课程的编号(例如 YW)放入隐藏表单元素,代码如 studentForm.jsp 所示。

studentForm.jsp

```

...
请您输入张海的语文成绩(可修改):
<form action="目标页面路径" method="post">
  输入成绩: <input type="text" name="score" >
    <input type="hidden" name="stuno" value="0015" >
    <input type="hidden" name="courseno" value="YW" >
  <input type="submit" value="修改">
</form >
...

```

这样,目标页面就可以在得知成绩的同时还得知该成绩所对应学生的学号和课程编号。

5.5 其他问题

5.5.1 用 JavaScript 进行提交

有时候可能要对表单中的输入进行一些验证,例如在登录表单中需要输入的账号、密码不能为空,因此在单击“提交”按钮时不能马上提交,应该调用 JavaScript 进行验证,然后进行提交,所以“提交”按钮的类型不能被设置为 submit,而应该设置为 button,代码如 jsSubmit.jsp 所示。

jsSubmit.jsp

```

<%@ page language="java" contentType="text/html; charset=gb2312"%>
<html >
  <body >
    欢迎登录学生管理系统
    <script type="text/javascript">
      function validate(){
        if(loginForm.account.value==""){
          alert("账号不能为空!");
          return;
        }
      }
    </script>
  </body >
</html >

```

```
    }
    if(loginForm.password.value == ""){
        alert("密码不能为空!");
        return;
    }
    loginForm.submit();
}
</script>
<form name = "loginForm" action = "target.jsp" method = "post">
    请您输入账号: <input name = "account" type = "text"><br>
    请您输入密码: <input name = "password" type = "password"><br>
    <input type = "button" value = "登录" onClick = "validate()">
</form>
</body>
</html>
```

运行,输入密码,账号为空,然后单击“登录”按钮,效果如图 5-29 所示。

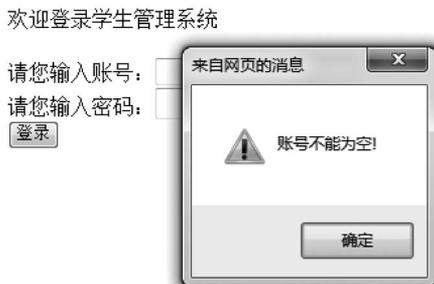


图 5-29 账号为空界面

如果将账号和密码都输入,系统会跳到 target.jsp。此处省略 target.jsp 的代码。

5.5.2 中文乱码问题

如果用户使用的是 Tomcat 服务器,在提交过程中经常会出现中文乱码问题,这在前面的章节中曾经提到过。

这里从两个方面讲解中文显示问题。

1. 中文无法显示

在有些 JSP 中,中文根本无法显示,通常的原因是没有把文件头上的字符集设置为中文字符集。用户一定要保证在文件头上写明如下内容。

```
<%@ page language = "java" contentType = "text/html; charset = gb2312" %>
```

或者:

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

2. 在提交过程中显示乱码

在 5.2.1 节提交“罗斯”出现了乱码,这是因为将“罗斯”提交给服务器时服务器将其认成 ISO-8859-1 编码,而网页上显示的是 gb2312 编码,两者不能兼容。通常有 3 种方法解决

这个问题。

(1) 将其转成 gb2312 格式,方法如下。

```
...
<%
    String stuname = request.getParameter("stuname");
    stuname = new String(stuname.getBytes("ISO-8859-1"),"gb2312");
    ...
%>
...
```

但是此种方法必须对每一个字符串进行转码,很麻烦。

(2) 直接修改 request 的编码。用户可以将 request 的编码修改为支持中文的编码,这样整个页面中的请求都可以自动转为中文,方法如下。

```
...
<%
    request.setCharacterEncoding("gb2312");
    String stuname = request.getParameter("stuname");
    ...
%>
...
```

用户一定要注意,该方法要在取出值之前设置 request 的编码,并且表单的提交方式应该是 post。但是,此种方法必须在每个页面中进行 request 的设置,也很麻烦。

(3) 利用过滤器。利用过滤器可以对整个 Web 应用进行统一的编码过滤,比较方便。该内容将在后面的章节中讲解。

本章小结

本章讲解了 JSP 编程中的表单开发,首先对表单的基本结构和基本属性进行了学习,然后学习了各种表单元素与服务器的交互,最后对隐藏表单的作用进行了介绍。

课后习题

一、填空题

1. 表单元素放在_____标签之间。
2. 表单元素提交给服务器端的哪个页面可以用< form >中的_____属性决定。
3. 在表单中提交请求的两种常见方式是_____和_____,默认情况下是_____方式。
4. 捆绑表单元素数据的获取方法为_____。
5. 将 type 属性设置为_____可以隐藏表单元素。

- 直接修改 request 的编码方式来解决中文乱码问题的代码是_____。
- HTTP 是无状态的协议,在页面之间传递值时必须通过_____。
- 在用 JavaScript 对表单的输入进行验证时,需要将“提交”按钮的类型设置为_____。

二、选择题

- 下列关于表单的说法不正确的是()。
 - 表单是可以由用户输入并提交给客户端的一个图形界面
 - 在表单中一般有一个按钮负责提交
 - 单击“提交”按钮,表单元素中的内容会提交给服务器端
 - 在表单中可以输入一些内容,这些输入功能由表单元素提供
- 有下面两段代码:

page1. jsp

```
<form action = "page2. jsp">
    请您输入账号: <input name = "account" type = "text"><br>
    请您输入密码: <input name = "password" type = "password"><br>
    <input type = "submit" value = "登录">
</form>
```

page2. jsp

```
<%
//获得表单中 name = account 的表单元素中输入的值,赋值给 str
String str1 = request.getParameter("account");
String str2 = request.getParameter("zhanghu");
%>
```

- 若不输入数据,直接单击“登录”按钮,则 str1 和 str2 的值分别是()。
- null、null
 - null、“”
 - “”、“”
 - “”、null
- 要想在浏览器的地址栏上的 URL 中隐藏传输的参数,需要将< form >标签的 method 属性值设置为()。
 - hidden
 - post
 - get
 - submit
 - 下列说法中错误的是()。
 - 除了复选框、多选列表框等,其他一些表单元素也可以设置为同名
 - 获取同名表单元素中数据的方法为 request.getParameterValues(arg)
 - 以文本框在源代码中出现的顺序从数组的第 0 位开始向后放置同名表单元素的数据
 - 在提交表单数据时只能用 post 方法
 - 下列表单元素中不是单一表单元素的是()。
 - 复选框
 - 文本框
 - 密码框
 - 单选按钮
 - 在表单提交的过程中,不能解决中文乱码问题的方案为()。
 - 将获取到的数据转换成 gb2312 的格式
 - 在获取数据之前先修改 request 的编码方式

- C. 把文件头上的字符集设置为中文字符集
 - D. 利用过滤器对整个 Web 应用进行统一的编码过滤
7. 下列关于用表单传值的说法正确的是()。
- A. 表单传值的数据可以是任何类型
 - B. 表单传值非常安全,即使是在客户端的源代码里也看不到传输的值
 - C. 将表单元素的 type 属性设置为"hidden",且将提交方式设置为 post 可完全隐藏传输的数据
 - D. 虽然表单传值的方法不是绝对安全的,但由于其简单性和平台支持的多样性,很多程序还是用表单传值

三、上机习题

1. 制作一个登录表单,输入账号和密码,如果账号、密码相符,则显示“登录成功”,否则显示“登录失败”。
2. 在上题的表单中增加一个 checkbox,让用户选择“是否注册为会员”,如果注册为会员,则在显示时增加文本“欢迎您注册为会员”。
3. 在页面 1 的表单内输入一个数字 N ,提交,能够在另一个页面打印 N 个“欢迎”字符串。
4. 编写一个“计算找零”的页面,在页面上输入应付款、实际付款,提交,在页面底部显示应该找零的数量和各种面额的张数,例如找零是 56 元,应该付款为 50 元 1 张、5 元 1 张、1 元 1 张。假设现有 50、20、10、5、1 这 5 种面额。
5. 在页面 1 中输入账号和密码,进行登录,如果账号和密码相符,则认为成功登录到页面 2,在页面 2 中显示一个文本框输入用户姓名,输入之后提交,在页面 3 中显示用户的账号和姓名。