HTML表单

王新阳
wxyyuppie@bjfu.edu.cn

主要内容

- HTML的表单和组件
- 与请求表单相关的主要API
- 表单上传文件的处理
- Servlet处理表单请求的细节问题

一、HTML的表单和组件

- 表单是具有输入域、文本域的页面
- 用于客户端同服务器端交互
- 用户在客户端填写表单，然后“提交”
- 表单中的信息发送到服务器
- 服务器接受表单信息并返回处理情况

一、HTML的表单和组件

- 表单主标记
  - **<form> </form>**
- 表单中的标记
  - **<input type=“”>**
  - **<select> </select>**
  - **<option>**
  - **<textarea> </textarea>**
- **<form>是表单开始标记，</form>结束**

一、HTML的表单和组件—form标记

- 描述表单的特性，语法

<form method="POST" action= " " enctype= " " >

- method表单传输方法
  - GET方法将表单信息在URL后传输
  - POST方法将表单信息作为信息体传输
- action为表单处理方式，通常为一个URL
- enctype为编码方式

一、HTML的表单和组件—form标记

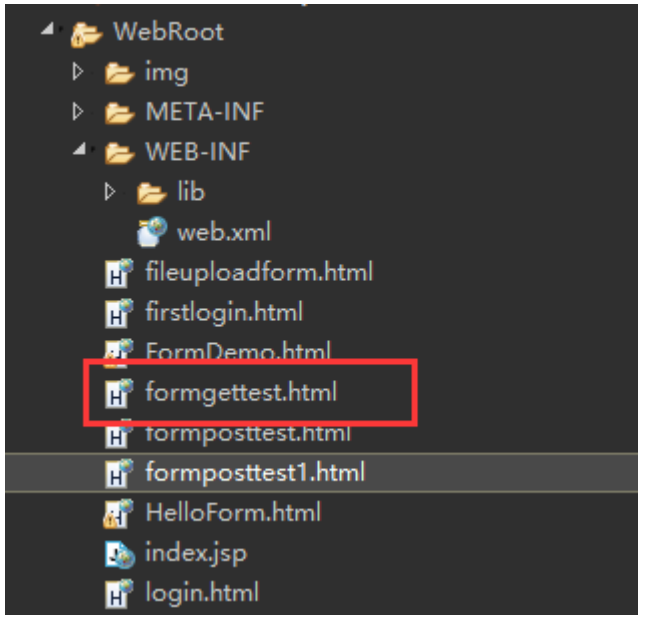
一个带表单的HTML文件

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
 <HTML>
  <head><title>表单简单示例</title></head>
  <body bgcolor="RED">
  <h2 align="center">使用Get方法的表单简单示例</h2>
  <form action="formtest.html" method="get">
      <center>姓名:
          <input type="text" name="name"  value="王芳"><br>
            学校:
          <input type="text" name="school" value="北京大学"><p>
          <input type="submit"> <!-- Press this to submit form -->
      </center>
  </form>
  </body>
</HTML>
```

一、HTML的表单和组件—form标记

- 标签不区分大小写
  - 在Tomcat的Web动态项目下，html文件必须放在WebContent（或WebRoot）目录下

- **<form> action属性**
  - 可以是相对路径
    - ✓ action=“*formactiontest.html*”
  - 可以是绝对路径
    - ✓ *action="http://localhost/ServletTest/formactiontest.html"*
  - 还可以是Servlet路径
    - ✓ action=“*ServletTest*”



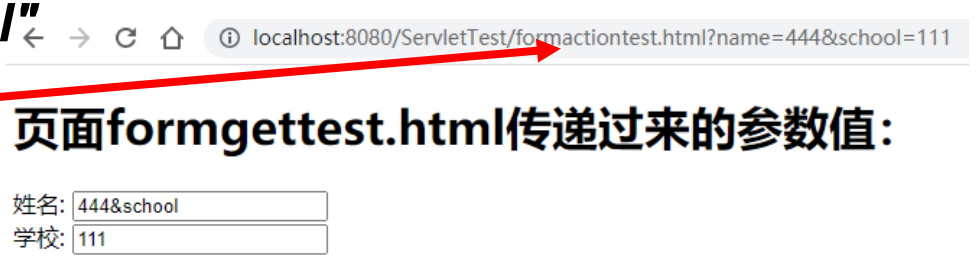
见示例ServletTest/formgettest.html

一、HTML的表单和组件—form标记

action=“*formactiontest.html*”

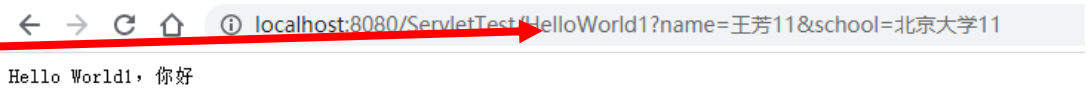
action=“*http://localhost:8080/ServletTest/formactiontest.html*"

```
<BODY BGCOLOR="#FDF5E6">
    <H2 ALIGN="CENTER">使用Get方法的表单简单示例</H2>
    <FORM ACTION="http://localhost/ServletTest/formactiontest.html"  method="get">
    <!--FORM ACTION="GetNameandSchool" method="get"-->
        <CENTER>
            姓名: <INPUT TYPE="TEXT" NAME="name" VALUE="王芳11"><BR> 学校:
            <INPUT TYPE="TEXT" NAME="school" VALUE="北京大学11">
            <P>
                <INPUT TYPE="SUBMIT">
                <!-- Press this to submit form -->
        </CENTER>
    </FORM>
</BODY>
</HTML>
```

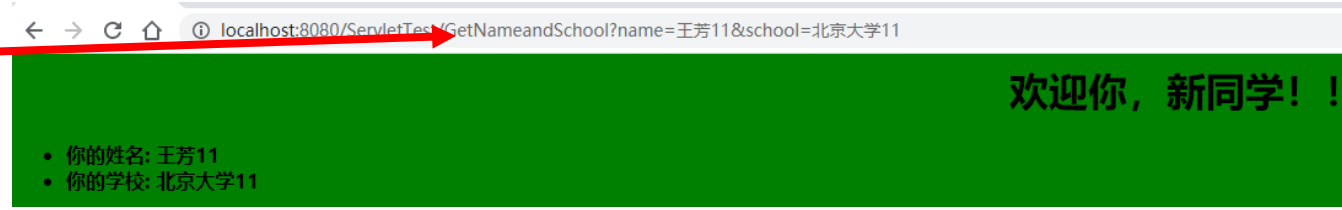


✓ **action=“*ServletName…*”**

```
<BODY BGCOLOR="#FDF5E6">
    <H2 ALIGN="CENTER">使用Get方法的表单简单示例</H2>
    <!--FORM ACTION="http://localhost/ServletTest/formactiontest.html"  method="get"-->
    <FORM ACTION="HelloWorld1" method="get">
        <CENTER>
            姓名: <INPUT TYPE="TEXT" NAME="name" VALUE="王芳11"><BR> 学校:
            <INPUT TYPE="TEXT" NAME="school" VALUE="北京大学11">
            <P>
```




```
<BODY BGCOLOR="#FDF5E6">
    <H2 ALIGN="CENTER">使用Get方法的表单简单示例</H2>
    <!--FORM ACTION="http://localhost/ServletTest/formactiontest.html"  method="get"-->
    <FORM ACTION="GetNameandSchool" method="get">
        <CENTER>
            姓名: <INPUT TYPE="TEXT" NAME="name" VALUE="王芳11"><BR> 学校:
            <INPUT TYPE="TEXT" NAME="school" VALUE="北京大学11">
            <P>
                <INPUT TYPE="SUBMIT">
                <!-- Press this to submit form -->
        </CENTER>
    </FORM>
</BODY>
```



一、HTML的表单和组件— <input>标记

- 表单中输入信息的区域
- 属性
  - **type** 类型
  - **name** 名称
  - **id** 标识
  - **maxlength** 最大字符数
  - **size** 输入域宽度
  - **value** 域的初始值
  - **button** 按钮 用**javascript**响应

<input>的类型-type

- text 文本
- passwowrd 口令方式
- checkbox 多选框 name相同
- radio 单选按钮 name相同
- image 图片
- hidden 隐藏表单 发送数据
- submit 提交按钮 向服务器提交表单
- reset 复位按钮 将表单充置为初始状态

一、HTML的表单和组件—文本框：TEXTAREA

- 多行文本域：<textarea> </textarea>
- 参数
  - COLS：宽度
  - ROWS：高度
  - NAME：名称

```
<textarea NAME="Computer" ROWS=6  COLS=64>
   CPU  PIV 1500
   Memory 512M
 </textarea>
```

一、HTML的表单和组件—下拉框：SELECT

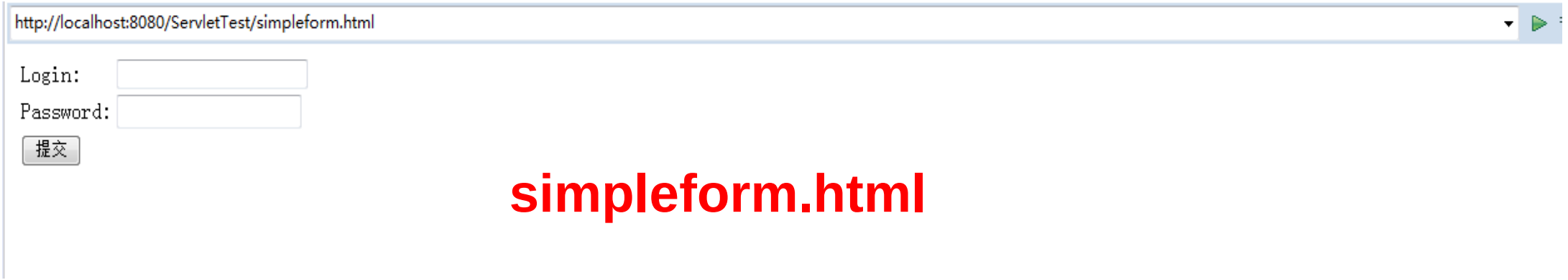
- 定义列表框：<select> </select>
- 参数
  - MULTIPLE 可同时选取多行
  - NAME 名称
  - SIZE 可见项目数
- 每个项目用option标记
- 属性SELECTED出现在option表示选中

```
<select name="address" size=1>
    <option value="sh">上海</option>
    <option value="bj" >北京</option>
    <option value="gd">广东</option>
</select>
```

一、HTML的表单和组件——表单示例1

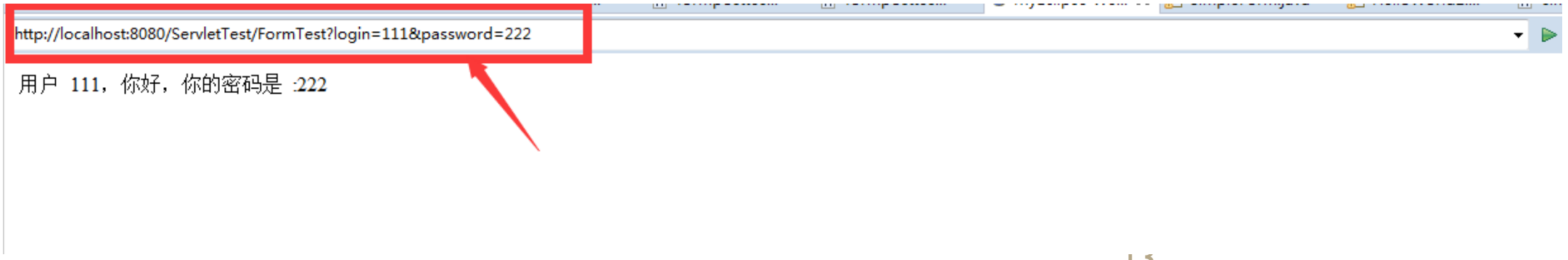
示例1——通过Web.xml配置

运行前



simpleform.html

运行后



一、HTML的表单和组件——表单示例1

simpleform.html

```
<html>
  <head>
    <title>FormDemo.html</title>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312">
  </head>

  <body>
    <!--form name="f1" id="f1" action="http://localhost:8080/ServletTest/servlet" method="get"-->

    <!-- 分别替换成如下代码，观察变化 -->
    <form name="f1" id="f1" action="FormTest" method="get">
    <!--form name="f1" id="f1" action="FormTest" method="post" --><!-- 只修改此处会报错，思考为什么？ -->
      <table>
        <tr>
          <td>Login:</td>
          <td><input type="text" name="login" id="login"></td>
        </tr>
        <tr>
          <td>Password:</td>
          <td><input type="password" name="password" id="password"></td>
        </tr>
        <tr>
          <td colspan="2"><input type="submit" value="提交"></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

表单简单示例1——通过Web.xml配置

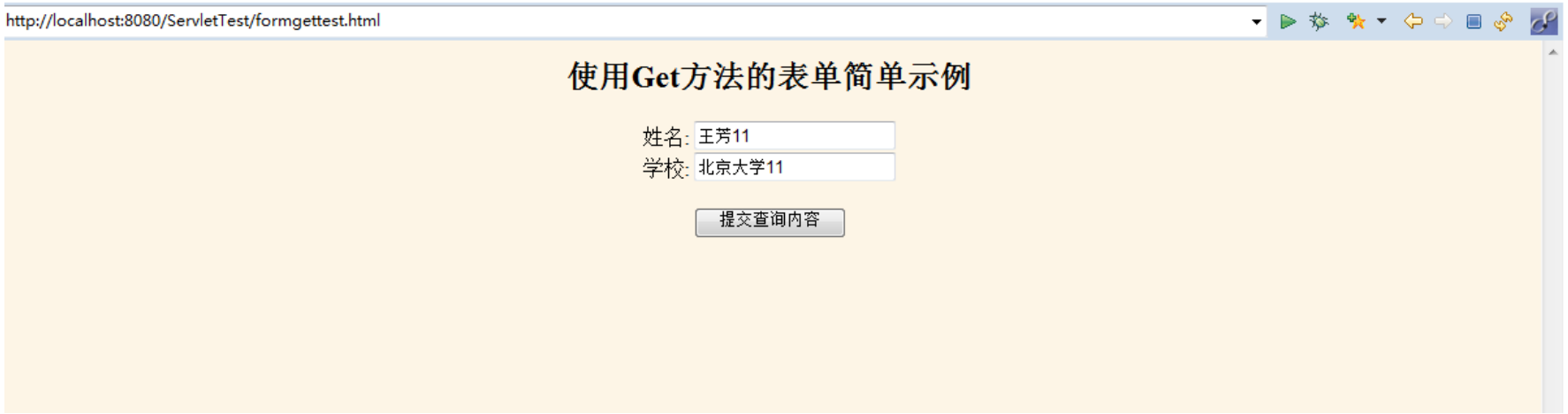
bjfu.FormTest.SimpleForm.java

```
public class SimpleForm extends HttpServlet {
public void doGet(HttpServletRequest request,HttpServletResponse response)
                                        throws ServletException, IOException {
//指定输出的编码格式，输出中文
      response.setCharacterEncoding("utf-8");
      response.setContentType("text/html;charset=UTF-8");
      String user=request.getParameter("login");
      String pwd=request.getParameter("password");
      PrintWriter out = response.getWriter();
      out.println("用户 "+user+"，你好，你的密码是 :"+pwd);
    }
}
```

一、HTML的表单和组件——表单示例2

表单简单示例2——get方法

运行前　　formgettest.html



运行后

表单简单示例2——get方法

formgettest.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<TITLE>Get表单简单示例</TITLE>
</HEAD>
<BODY BGCOLOR="#FDF5E6">
<H2 ALIGN="CENTER">使用Get方法的表单简单示例</H2>
<!--FORM ACTION="http://localhost/ServletTest/formtest.html"  method="get"-->
<FORM ACTION="GetNameandSchool" method="get">
<CENTER>
姓名: <INPUT TYPE="TEXT" NAME="name" VALUE="王芳11"><BR> 学校:
<INPUT TYPE="TEXT" NAME="school" VALUE="北京大学11">
<P>
<INPUT TYPE="SUBMIT">
<!-- Press this to submit form -->
</CENTER>
</FORM>
</BODY>
</HTML>
```

表单简单示例2——get方法

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    doProcessForm(request,response);

}

protected void doProcessForm(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {{
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    String docType =
        "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 "+
        "Transitional//EN\">\n";
    String title="欢迎你，新同学！！";
    out.println(docType +
            "<HTML>\n" +
            "<HEAD><TITLE>"+title + "</TITLE></HEAD>\n" +
            "<BODY BGCOLOR=\"green\">\n" +
            "<H1 ALIGN=\"CENTER\">" + title + "</H1>\n" +
            "<UL>\n" +
            " <LI><B>你的姓名:"
            //+new String(request.getParameter("name").getBytes("ISO-8859-1"),"GBK")+ "\n" +
            +request.getParameter("name")+ "</B>\n" +
            " <LI><B>你的学校:"
            //+ new String(request.getParameter("school").getBytes("ISO-8859-1"),"GBK") + "\n" +
            +request.getParameter("school")+ "</B>\n" +
            "</UL>\n" +
            "</BODY></HTML>");
            }
}
```

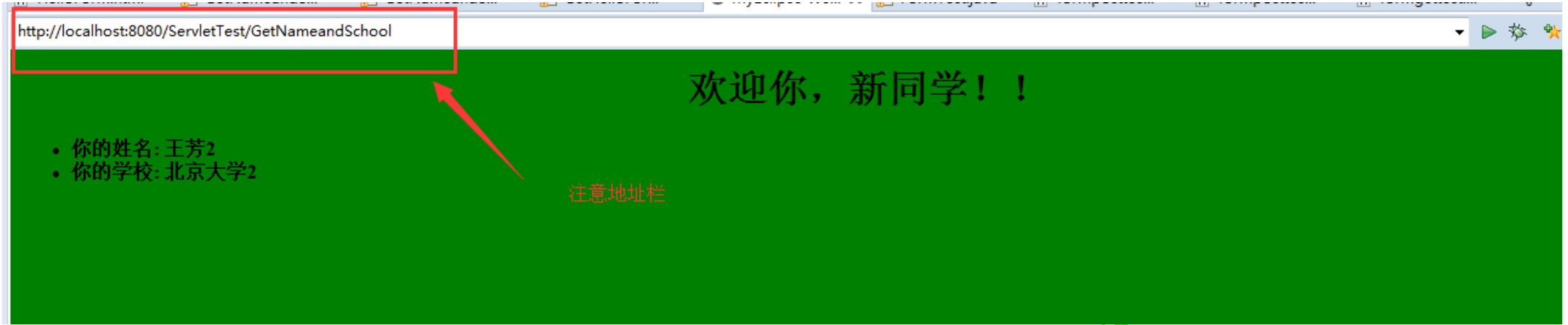
一、HTML的表单和组件——表单示例3

表单简单示例3——post方法

运行前

formposttest.html

运行后



表单简单示例3——post方法

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<TITLE>Post表单简单示例</TITLE>
</HEAD>
<BODY BGCOLOR="yellow">
<H2 ALIGN="CENTER">使用Post方法的表单简单示例</H2>
<!--FORM ACTION="http://localhost/ServletTest/formtest.html"  method="post"-->
<FORM ACTION="GetNameandSchool" method="post">
<CENTER>
姓名: <INPUT TYPE="TEXT" NAME="name" VALUE="王芳2"><BR> 学校:
<INPUT TYPE="TEXT" NAME="school" VALUE="北京大学2">
<P>
<INPUT TYPE="SUBMIT">
<!-- Press this to submit form -->
</CENTER>
</FORM>
</BODY>
</HTML>
```

表单简单示例3——post方法

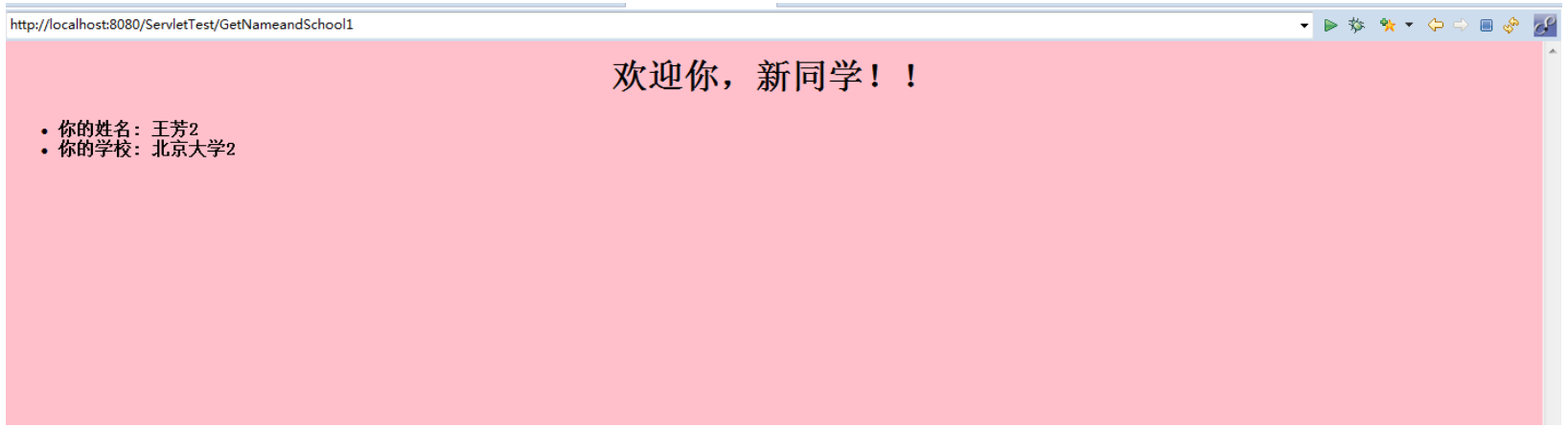
```
    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        //把Web服务器对请求Body部分的编码设为UTF-8，这样和输出编码一致，输出时就不必从ISO8859转换过来！
        //request设置的编码应该和前面表单网页设置的编码格式一致，这样才不会出现乱码！
        request.setCharacterEncoding("UTF-8");
        doProcessForm(request,response);
    }
```

调用同一个处理函数

一、HTML的表单和组件——表单示例3

单独实现post方法　　**formposttest1.html**

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    {
        response.setContentType("text/html;charset=GBK");
        PrintWriter out = response.getWriter();
        String docType =
            "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 "+
            "Transitional//EN\">\n";
        String title="欢迎你，新同学！！";
        out.println(docType +
                "<HTML>\n" +
                "<HEAD><TITLE>"+title + "</TITLE></HEAD>\n" +
                "<BODY BGCOLOR=\"pink\">\n" +
                "<H1 ALIGN=\"CENTER\">" + title + "</H1>\n" +
                "<UL>\n" +
                " <LI><B>你的姓名:"
                //注意，这里参数的名称应该和表单中的元素名字大小写一致
                +new String(request.getParameter("name").getBytes("ISO-8859-1"),"GBK")+ "\n" +
                " <LI><B>你的学校:"
                + new String(request.getParameter("school").getBytes("ISO-8859-1"),"GBK") + "\n" +
                "</UL>\n" +
                "</BODY></HTML>");
                }
}
```



表单综合示例——HelloForm.html

用户注册

用户名： 密码：

你喜欢： 足球 篮球 性别： 男 女

你的计算机

```
CPU  PIV 1500
Memory 512M
```

你计算机的操作系统 

你所在地： 天津

提交 全部重写

```java
@WebServlet("/GetHelloForm")
public class GetHelloForm extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public GetHelloForm() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        String formHead="<HTML>\n" +
        "<HEAD><TITLE>"+ "读取请求表单的各种元素" + "</TITLE></HEAD>\n" +
        "<BODY BGCOLOR=\"#FDF5E6\">\n" +
        "<H1 ALIGN=CENTER>" + "读取注册信息" + "</H1>\n" +
        "<TABLE BORDER=1 ALIGN=CENTER>\n" +
        "<TR BGCOLOR=\"#FFAD00\">\n" +
        "<TH>Parameter Name<TH>Parameter Value(s)";

        String formEnd="</TABLE>\n</body></html>";

        out.print(formHead);

        Enumeration paramNames = request.getParameterNames();
        while(paramNames.hasMoreElements()) {
          String paramName = (String)paramNames.nextElement();
          out.print("<TR><TD>" + paramName + "\n<TD>");
          String[] paramValues =
            request.getParameterValues(paramName);
```

```
<form method="post" action="GetHelloForm">
<p align="center">用户注册

<p align="center">
用户名：<input type="text" name="User" size="20">    
密码：<input type="password" name="pwd" size="20"><br> <br>

<p align="center">你喜欢：
<input type="checkbox" name="sports" value=football> 足球
<input type="checkbox" name="sports" value=bastketball> 篮球

    
性别：
<input type="radio" name="sexy" value=male> 男
<input type="radio" name="sexy" value=female> 女 <br><br>

<p align="center">你的计算机
<textarea NAME="Computer" ROWS=6  COLS=64>
    CPU  PIV 1500
    Memory 512M
 </textarea><br><br>
<p align="center">你计算机的操作系统
<input type="image" name="os" src="img\window7.jpg" width="40" height="30">

    
<p align="center">你所在地：

<select  name="area" width="50"  size="1">
    <option value="bj"  selected > 北京 </option>
    <option value="tj" > 天津 </option>
    <option value="sh" > 上海 </option>
    <option value="cq" > 重庆 </option>
</select>

<br> <br>
 <input type="submit" value=" 提交">
 <input type="reset" value="全部重写"></p>
</form>
```

表单综合实例——HelloForm.html

- 表单
  - <form method="POST" action="GetFormTest2" >
  - //注意这时表单的enctype属性默认为application/x-www-form-urlencoded
- 表单元素包括
  - <input type="text"…
  - <input type="password" …
  - <input type="checkbox" …
  - <input type="radio"…
  - <input type="image"…
  - <textarea …
  - <select …

表单综合实例——HelloForm.html

语法

```
<form enctype="value">
```

属性值

值	描述
application/x-www-form-urlencoded	在发送前编码所有字符（默认）
multipart/form-data	不对字符编码。 在使用包含文件上传控件的表单时，必须使用该值。
text/plain	空格转换为 "+" 加号，但不对特殊字符编码。

一、HTML的表单和组件——综合示例

一种简单的servlet注册方式——基于注释

```
<form method="post" action="GetHelloForm">
<p align="center">用户注册

<p align="center">
用户名：<input type="text" name="User" size="20">    
密码：<input type="password" name="pwd" size="20"><br> <br>

<p align="center">你喜欢：
<input type="checkbox" name="sports" value=football> 足球
<input type="checkbox" name="sports" value=bastketball> 篮球

    
性别：
<input type="radio" name="sexy" value=male> 男
<input type="radio" name="sexy" value=female> 女 <br><br>
```

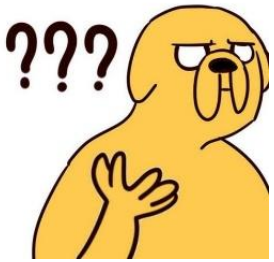
```
import java.io.IOException;

/**
 * Servlet implementation class GetFormTest2
 */
@WebServlet("/GetHelloForm")
public class GetHelloForm extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public GetHelloForm() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
```

在web.xml中没有与之对应的servlet？




```
  <welcome-file>index.jsp</welcome-file>
  <welcome-file>default.html</welcome-file>
  <welcome-file>default.htm</welcome-file>
  <welcome-file>default.jsp</welcome-file>
</welcome-file-list>
<servlet>
  <description></description>
  <servlet-name>HelloWorld1</servlet-name>
  <servlet-class>bjfu.SimpleServelet.HelloWorld1</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>HelloWorld1</servlet-name>
  <url-pattern>/HelloWorld1</url-pattern>
</servlet-mapping>


<servlet>
  <description></description>
  <servlet-name>HelloWorld2</servlet-name>
  <servlet-class>bjfu.SimpleServelet.HelloWorld2</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>HelloWorld2</servlet-name>
  <url-pattern>/HelloWorld2</url-pattern>
</servlet-mapping>


<servlet>
  <description></description>
  <servlet-name>FormTest</servlet-name>
  <servlet-class>bjfu.FormTest.SimpleForm</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>FormTest</servlet-name>
  <url-pattern>/FormTest</url-pattern>
</servlet-mapping>
```

GetHelloForm.java

- 处理一系列表单元素

```
//取表单元素名称放到一个数组
Enumeration paramNames =
  request.getParameterNames();

 while(paramNames.hasMoreElements()) {
  //对每一个表单元素，取其名称
   String paramName =
(String)paramNames.nextElement();
  //输出表单元素名称
   out.print("<TR><TD>" + paramName + "\n<TD>");
   //获得表单元素的一系列值
  String[] paramValues =
    request.getParameterValues(paramName);
```

一、HTML的表单和组件——综合示例

GetHelloForm.java

```
//判断表单元素是否只有一个值
if (paramValues.length == 1) {
    String paramValue = paramValues[0];
  //判断表单元素值是否为null
    if (paramValue.length() == 0)
      out.println("<I>No Value</I>");
    else
      out.println(paramValue);
  } else {//若表单元素不只有一个值
    out.println("<UL>");
    for(int i=0; i<paramValues.length; i++) {
      out.println("<LI>" + paramValues[i]);
    }
    out.println("</UL>");
  }   }
```

二、与请求表单相关的主要API

- 获取表单请求参数的核心API：
  - **String** request.getParameter("参数名")：根据参数名获取参数值（注意，只能获取一个值的参数）
  - 例如
    - ✓ **String xx=request.getParameter("school");**
  - 注意，这里参数的名称应该和表单中的元素名字大小写一致
    - **Enumeration** request.getParameterValue("参数名 ")：根据参数名获取参数值（可以获取多个值的参数）
    - **Enumeration** request.getParameterNames()：获取所有参数名称列表

二、与请求表单相关的主要API

HttpServletRequest request

- 是浏览器向服务器端发送的请求对象，通过该对象获取请求数据
- 核心的API：
  - 请求行：
    - request.getMethod()；获取请求方式
    - request.getRequetURI()　/ request.getRequetURL()　请求资源信息
    - request.getProtocol()　请求http协议版本
  - 请求头：
    - request.getHeader("名称")　根据请求头获取请求值
    - request.getHeaderNames()　获取所有的请求头名称

二、与请求表单相关的主要API

HttpServletResponse response

1. 对响应数据进行封装的对象

2. 输出数据

- 1)getOutputStream字节输出流
  - response.getOutputStream().write("中国".getBytes("utf-8"));
- 2）getWriter字符输出流
  - response.getWriter().write("北京");
  - 注意:
    - 以上两种形式不能混搭在一起
    - 用默认的ISO-8859-1对数据编码(该码表中没有汉字，因此汉字会被编码为?传送到浏览器上).

二、与请求表单相关的主要API

3. 设置相应的头部信息

- 1）解决乱码
  - response.setCharacterEncoding("utf-8");
- 2）通知浏览器接受数据时使用utf-8解码
  - response.setHeader("Content-Type", "text/html;charset=utf-8");
- 3）以上两者合二为一
  - response.setContentType("text/html;charset=utf-8");
  - 建议同时设置 characterEncoding 和 Content-Type.
- 4）实现页面的自动刷新和跳转
  - response.setHeader(“Refresh”, “3;url=HelloForm.html");

二、与请求表单相关的主要API

处理Enumeration数据

- 示例 **ServletTest.war**
  - ✓**HelloForm.html**
  - ✓**bjfu.FormTest.GetHelloForm.java**

```
Enumeration paramNames=request.getParameterNames();
while(paramNames.hasMoreElements()) {
    String paramName=(String)paramNames.nextElement();
        …
}
```

二、与请求表单相关的主要API

Iterator和Enumeration的区别

- java中的集合类都提供了返回Iterator的方法，就是迭代器，它和Enumeration（枚举）的主要区别其实就是Iterator可以删除元素，但是Enumration却不能。

- Enumeration 接口的功能与 Iterator 接口的功能是重复的。只是，Iterator 接口添加了一个可选的移除操作，并使用较短的方法名。新的实现应该优先考虑使用 Iterator 接口而不是 Enumeration 接口。

Iterator的基本方法

```
public interface Iterator {
        boolean hasNext();
        Object next();
        void remove();
    }
```

二、与请求表单相关的主要API

✓**Enumeration向量数组**

```
Enumeration paramNames=request.getParameterNames();
while(paramNames.hasMoreElements()) {
    String paramName =(String)paramNames.nextElement();
                        …
}
```

✓**迭代器数组（List →iterator）**

```
//解析表单中提交的所有文件内容
  items=upload.parseRequest(request).iterator();
  while(items.hasNext()){
      FileItem item = (FileItem) items.next();
                        ...
  }
```

三、表单上传文件的处理

- 1. 上传文件表单的设置
- 2. Servlet的处理
  - 方法1：利用cos.jar
  - 方法2：利用commons-fileupload.jar
  - 原因：表单中的文件上传时编码为二进制，需要第三方组件特殊处理

比较常见的文件上传组件有Commons FileUpload（http://jakarta.apache.org/commons/fileupload>）和COS FileUpload（http://www.servlets.com/cos）

三、表单上传文件的处理—fileuploadform.html

- 上传文件表单的注意事项
  - **<form method="POST" action="GetFormTest2" ENCTYPE="multipart/form-data">**
  - **注意1:这时表单的方法一定是post，并且enctype属性一定要指定为 multipart/form-data,才能上传文件**
  - **注意2： ENCTYPE的编码形式是按ISO-8859-1编码，导致输出中文表单值为乱码，因此需要在输出前对其转换，采用new String(a.getBytes("ISO-8859-1"),"UTF-8")**

- 表单元素可以包括
  - **<input type="file"…**
  - **<input type="text"…**
  - **<input type="password" …**
  - **<input type="checkbox" …**
  - **<input type="radio"…**
  - **<input type="image"…**
  - **<textarea …**
  - **<select …**

&&

```
<INPUT TYPE="file"
NAME="uploadfile" ></p>
```

三、表单上传文件的处理—fileuploadform.html

GetFileByCos.java

- 处理一系列表单元素
  - 在lib库中引入cos.jar
    1. 在Servlet类中
       import com.oreilly.servlet.MultipartRequest;
    2. 把request转换为MultipartRequest
       MultipartRequest request1=new MultipartRequest(request, “c:\\”);//其中“c:\\”表示上传文件存储的位置
    3. 获得表单所有元素名称
       Enumeration request1.getParameterNames()
    4. 获得一个表单元素的所有取值
       Enumeration request1.getParameterValues(String name)
    5. 添加一个Sting toUTF8(String)函数
    6. 输出参数时统一加上toUTF8函数，out.println(toUTF8(paramValue));
- 示例 ServletTest.war
  - ✓ fileuploadform.html
  - ✓ bjfu.FileUpload.GetFileByCos.java
- 特点：
  - 除了声明目标文件夹之外，不需要特别处理文件，使用方式类似原有的request

三、表单上传文件的处理—使用commons-fileupload.jar

1. 引入两个jar包 commons-fileupload.jar commons-io.jar
2. 构建文件上传处理对象，并将request手工包装为一个数据迭代器

```
FileItemFactory factory = new DiskFileItemFactory();
ServletFileUpload upload = new ServletFileUpload(factory);
Iterator  items=upload.parseRequest(request).iterator();
```

3. 依次处理迭代器中的每个数据，处理前先判断是否为正常表单域
   若为正常获取。输出表单域的值即可
4. 对文件域，需要手工获得文件名称后上传文件

示例 ServletTest.war

- ✓ **fileuploadform.html**
- ✓ bjfu.FileUpload.GetFileByCommons.java

四、Servlet处理表单请求的细节问题

1. 如何对数据缺失或异常进行检查？
2. 如何处理出现的数据缺失或异常？
3. Servlet中如何输出特殊字符？

四、Servlet处理表单请求的细节问题

1. 如何对数据缺失或异常进行检查

- 缺失
  - 表单中缺失某个字段
    - getParameter返回null，例如没有选择性别(其它元素)
  - 表单提交时字段为空
    - getParameter返回空字符串（或者由空格组成的字符串），例如，没有输入用户名或密码(text元素)
- 异常
  - 有输入，但不符合输入格式要求
    - 需要根据输入格式的具体要求进行检查

四、Servlet处理表单请求的细节问题

1. 如何对数据缺失或异常进行检查

- 检查数据缺失
  - 添加检查函数
    - ✓protected boolean isParameterNull(String p)
    - ✓protected boolean isParameterBlank(String p)
  - 使用参数值前进行检查
    - ✓注意在检查字符串是否为空之前必须检查它是否为null

```
String pwd=request.getParameter("PWD");
 if (isParameterNull(pwd))
    out.println("<p><b>null</b>\n");
  else if (isParameterBlank(pwd))
   out.println("<p><b>blank</b>\n");
  else
  out.println("<p><b>"+pwd+"</b>\n");
```

- 示例 **ServletTest.war**
  - ✓missingtest.html
  - ✓GetMissingTest.java

四、Servlet处理表单请求的细节问题

2. 如何处理出现的数据缺失或异常?

- 使用默认值替代缺失的值
- 再次显示表单
  - **显示原来的空表单**
    - ✓response.sendRedirect("missingtestwithdefault.html");
    - ✓或者

```
//不允许有之前有response.getWriter()语句，否则出现乱码
import javax.servlet.RequestDispatcher;
RequestDispatcher
   rd=request.getRequestDispatcher("missingtestwithdefault.html");
    rd.forward(request, response);
```

  - **再次显示表单，将缺失的值标示出来, 之前已经输入的值应该保留—需要使用jsp文件再现表单**
- 示例 **ServletTest.war**
  - ✓**missingtestwithdefault.html**
  - ✓**GetMissingTestWithDefault.java**

四、Servlet处理表单请求的细节问题

3. Servlet中如何输出特殊字符?

- 特殊字符
  - <和>在任何地方都会引起问题
  - &和“在HTML属性中会引发问题
  - ≤和≥等特殊符号
- HTML中特殊符号——采用特别的编码
  - <（左尖括号、小于号） <
  - >（右尖括号、大于号） >
  - ≤ ≤
  - ≥ ≥
  - 全部请参见：http://www.002pc.com/master/College/Page/HTML/2008-10-06/1759.html
- 有时手动的转换是不可能的
  - 字符串来源于HTML表单数据或其它程序

四、Servlet处理表单请求的细节问题

3. Servlet中如何输出特殊字符?

- 把特殊字符进行转换
  - 增加一个函数判断输入是否有特殊字符
    - ✓**boolean hasSpecialChars(String input)**
  - 增加一个函数把特殊字符进行转换
    - ✓**String filter(String input)**
  - 对需要处理的字符串进行转换
    - **//必须加上PRE， PRE元素内的文字是已经格式化的。空格和回车都是预留的。**
    - **out.println("<PRE>\n" +*filter*(request.getParameter("intro"))+"</PRE>\n");**
- 示例 **ServletTest.war**
  - ✓**specialchartest.html**
  - ✓**GetSpecialCharTest.java**

四、Servlet处理表单请求的细节问题

4. 中文乱码问题

- Tomcat服务器解析请求表单时的编码规则是：
  - URL——UTF8
  - Body——ISO-8859-1
- Get方法，参数在URL中
  - 浏览器的UTF8 →服务器的UTF8→ response为UTF8
- Post方法，参数在body中
  - 浏览器的UTF8→服务器的ISO-88591 →那么需要request为UTF8→response为UTF8

中文表单提交和处理使用总结

- 请求表单的HTML文件(不论表单采用post还是get方法提交)
  - <head>…</head>中增加
    - <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
- 处理表单的Servlet文件
  - doGet方法的最前面添加：
    - response.setContentType("text/html;charset=UTF-8");
  - doPost方法的最前面添加：
    - request.setCharacterEncoding(“UTF-8”);//输入信息统一为UTF-8编码
    - response.setContentType("text/html;charset=UTF-8");
    - 或者省略request.setCharacterEncoding的设置，对接受的每个数据进行重编码工作：
    - new String(request.getParameter("school").getBytes("ISO-8859-1"),"utf-8")

练习——实验1的主要内容

- 设计一个注册表单名为register.html
  - 请用户输入用户名和密码
- 设计一个处理注册表单的Servlet，名为GetReg.java
  - 检查用户输入的用户名，若为aaa,则重新回到register.html，请用户重新输入；否则进入resume.html页面，请用户输入简历信息
- 设计一个简历表单名为resume.html
  - 要求有姓名\出生日期\学校\简介等信息输入
  - 要求包括所有的表单元素(pwd可以忽略)
- 设计一个处理简历表单的Servlet，名为GetResume.java
  - 要求把用户的简历信息以表格的样式显示出来
  - 要有一定处理特殊字符和数据缺失异常的处理

感谢聆听

Thanks For Your Listening!