**javaweb模块级开发示例**

目录

[1、html+css+js示例 3](#_Toc5299)

[2、 表单接收示例 5](#_Toc1445)

[Servlet 5](#_Toc26362)

[Jsp 5](#_Toc43)

[3、JavaBeans示例 7](#_Toc12282)

[4、会话管理 9](#_Toc21917)

[使用HttpSession对象管理会话。 10](#_Toc4084)

[使用Cookie跟踪会话 11](#_Toc16349)

[5、 数据库访问 15](#_Toc12613)

# 1、html+css+js示例

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>网页示例</title>

<style>

body {

background-color:#d0e4fe;

}

h1 {

color:orange;

text-align:center;

}

p {

font-family:"Times New Roman";

font-size:20px;

}

</style>

<script>

function formSubmit(){

var x = document.forms["frm1"]["username"].value;

if (x == null || x == "") {

alert("需要输入名字。");

return false;}

else{

document.getElementById("frm1").submit();

}

}

</script>

</head>

<body>

<h1>CSS 实例：这是一个标题!</h1>

<p>这是一个段落。</p>

<form id="frm1" action="form\_action.jsp">

用户名称: <input type="text" name="username"><br>

登录密码: <input type="text" name="passwd" required="required"><br><br>

<input type="button" onclick="formSubmit()" value="提交表单">

</form>

</body>

</html>

# 表单接收示例

## Servlet

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class formshow1

\*/

@WebServlet("/formshow1")

**public** **class** formshow1 **extends** HttpServlet {

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** **void** service(HttpServletRequest request,HttpServletResponse response)

**throws** ServletException, IOException{

String userName = request.getParameter("uName");

PrintWriter pw = response.getWriter();

pw.println("<html>");

pw.println("<head>");

pw.println("</head>");

pw.println("<body>");

pw.println("<h3>Hello, " + userName + "</h3>");

pw.println("<h3>Hello, " + **new** java.util.Date() + "</h3>");

pw.println("</body>");

pw.println("</html>");

}

}

## Jsp

<html>

<head>

<title>Using GET and POST Method to Read Form Data</title>

</head>

<body>

<center>

<h1>Read Form Data</h1>

<ul>

<li><p><b>user Name:</b>

<%= request.getParameter("uname")%>

</p></li>

<li><p><b>password:</b>

<%= request.getParameter("passwd")%>

</p></li>

</ul>

</body>

</html>

# 3、JavaBeans示例

创建一个JavaBean包，并在其下创建StudentBean的class。下面是一个属性很少的StudentBean.java 文件：

public class StudentBean implements java.io.Serializable {

private String firstName = null;

private String lastName = null;

private int age = 0;

public void StudentsBean() {

}

public String getFirstName(){

return firstName;

}

public String getLastName(){

return lastName;

}

public int getAge(){

return age;

}

public void setFirstName(String firstName){

this.firstName = firstName;

}

public void setLastName(String lastName){

this.lastName = lastName;

}

public void setAge(int age){

this.age = age;

}

}

## 访问JavaBean

<jsp:useBean> 标签可以在 JSP 中声明一个 JavaBean，然后使用。声明后，JavaBean 对象就成了脚本变量，可以通过脚本元素或其他自定义标签来访问。

[jsp:useBean](http://222.30.145.84/admin/courseAdmin/expAdmin/guides/jsp:useBean" \t "http://222.30.145.84/admin/courseAdmin/expAdmin/guides/_blank) 标签的语法格式如下：

<jsp:useBean id = "bean's name" scope = "bean's scope" typeSpec></jsp:useBean>

其中，根据具体情况，scope 的值可以是 page，request，session 或 application。id值可任意只要不和同一 JSP 文件中其它<jsp:useBean>中 id 值一样就行了。

## 访问 JavaBean 对象的属性

在 <jsp:useBean> 标签主体中使用 <jsp:getProperty></jsp:getProperty> 标签来调用 getter 方法，使用 <jsp:setProperty></jsp:setProperty> 标签来调用 setter 方法，语法格式如下：

<jsp:useBean id = "id" class = "bean's class" scope = "bean's scope">

<jsp:setProperty name = "bean's id" property = "property name"

value = "value"></jsp:setProperty>

<jsp:getProperty name = "bean's id" property = "property name"></jsp:getProperty>

...........

</jsp:useBean>

name属性指的是Bean的id属性。property属性指的是想要调用的getter或setter方法。

接下来给出使用以上语法进行属性访问的一个简单例子：

<html>

<head>

<title>get and set properties Example</title>

</head>

<body>

<jsp:useBean id = "students" class = "JavaBean.StudentsBean">

<jsp:setProperty name = "students" property = "firstName" value = "Zara"></jsp:setProperty>

<jsp:setProperty name = "students" property = "lastName" value = "Ali"></jsp:setProperty>

<jsp:setProperty name = "students" property = "age" value = "10"></jsp:setProperty>

</jsp:useBean>

<p>Student First Name:

<jsp:getProperty name = "students" property = "firstName"></jsp:getProperty>

</p>

<p>Student Last Name:

<jsp:getProperty name = "students" property = "lastName"></jsp:getProperty>

</p>

<p>Student Age:

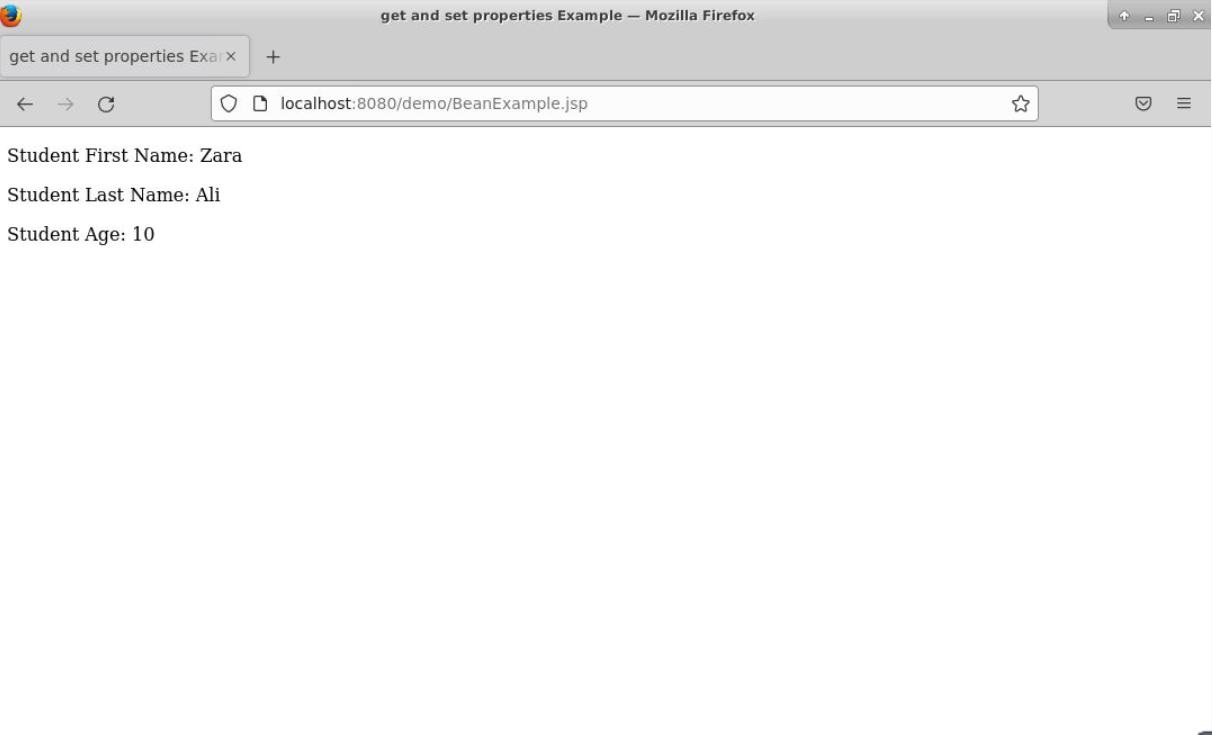
<jsp:getProperty name = "students" property = "age"></jsp:getProperty>

</p>

</body>

</html>

将以上代码放入BeanExample.jsp中。运行结果如下：



# 4、会话管理

Web服务器跟踪客户的状态方法通常有：

（1）使用HttpSession对象管理会话；（2）使用持久的Cookie对象；

## 使用HttpSession对象管理会话。

新建一个名为ShowSession的Servlet类，代码如下

package com.demo;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.util.Date;

public class ShowSession extends HttpServlet{

public void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

HttpSession session = request.getSession();

String heading =null;

Integer accessCount = (Integer)session.getAttribute("accessCount");

if(accessCount==null){

accessCount = new Integer(0);

heading = "Welcome, Newcomer";

}else{

heading = "Welcome Back.";

accessCount = new Integer(accessCount.intValue()+1);

}

session.setAttribute("accessCount",accessCount);

PrintWriter out = response.getWriter();

out.println("<html><head>");

out.println("<title>Session Tracking Example</title></head>");

out.println("<body><center>");

out.println("<h1>"+heading+"</h1>");

out.println("<h2>Information about your session.</h2>");

out.println("<table border='1'>");

out.println("<tr bgcolor=\"ffad00\"><th>Info Type<th>Value\n");

out.println("<tr><td>ID<td>"+session.getId()+"\n");

out.println("<tr><td>Creation Time:<td>");

out.println(""+new Date(session.getCreationTime())+"\n");

out.println("<tr><td>Time of last access:<td>");

out.println(""+new Date(session.getLastAccessedTime())+"\n");

out.println("<tr><td>Access number:<td>"+accessCount+"\n");

out.println("</table></center></body></html>");

}

}

在web.xml中加入内容

<servlet>

<servlet-name>ShowSession</servlet-name>

<servlet-class>com.demo.ShowSession</servlet-class>

</servlet>

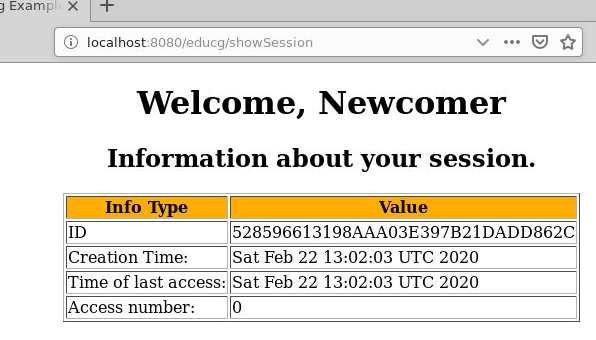
<servlet-mapping>

<servlet-name>ShowSession</servlet-name>

<url-pattern>/showSession</url-pattern>

</servlet-mapping>

运行ShowSession查看结果



## 使用Cookie跟踪会话

新建一个名为SendCookie的Sevlet类，向客户端写入Cookie对象

package com.demo;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class SendCookie extends HttpServlet{

public void doGet(HttpServletRequest request,

HttpServletResponse response)

throws IOException,ServletException{

Cookie userCookie = new Cookie("user", "uid1234");

userCookie. setMaxAge(60\*60\*24\*7);

response.addCookie(userCookie);

response.setContentType("text/html");

PrintWriter out=response.getWriter();

out.println("<html><title>add cookies</title>");

out.println("<body><h2>"+

"A cookie has been sent to brower</h2></body>");

out.println("</html>");

}

}

新建一个名为GetCookie的Servlet类，读取客户端的Cookie对象

package com.demo;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class GetCookie extends HttpServlet{

public void doGet(HttpServletRequest

request,HttpServletResponse response)

throws IOException,ServletException{

String cookieName = "user";

String cookieValue = null;

Cookie[] cookies = request.getCookies();

if (cookies!=null){

for(int i = 0;i<cookies.length;i++){

Cookie cookie = cookies[i];

if(cookie.getName().equals(cookieName))

cookieValue = cookie.getValue();

}

}

response.setContentType("text/html");

PrintWriter out=response.getWriter();

out.println("<html><title>get cookies</title>");

out.println("<body><h2>A cookie has been got from brower</h2>");

out.println("CookieName:"+cookieName+"<br>");

out.println("CookieValue:"+cookieValue+"<br>");

out.println("</body></html>");

}

}

在web.xml中添加内容

<servlet>

<servlet-name>SendCookie</servlet-name>

<servlet-class>com.demo.SendCookie</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>SendCookie</servlet-name>

<url-pattern>/sendCookie</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>GetCookie</servlet-name>

<servlet-class>com.demo.GetCookie</servlet-class>

</servlet>

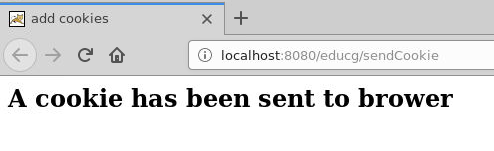
<servlet-mapping>

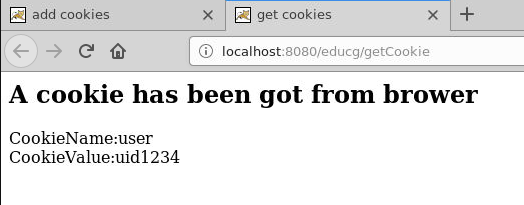
<servlet-name>GetCookie</servlet-name>

<url-pattern>/getCookie</url-pattern>

</servlet-mapping>

运行SendCookie和GetCookie查看结果

运行SendCookie向客户端写入一个Cookie  


运行GetCookie后可查看对应目录看是否写了一个文件  


# 数据库访问

数据库：TEST

数据表：Employees ，字段：id（int）, first（string）, last（string）, age（int）

用户：root 密码：1234

import java.io.\*;

import java.util.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.sql.\*;

public class DatabaseAccess extends HttpServlet{

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// JDBC driver name and database URL

final String JDBC\_DRIVER = "com.mysql.cj.jdbc.Driver";

final String DB\_URL="jdbc:mysql://localhost:3306/Employees?useUnicode=true&characterEncoding=utf8&serverTimezone=GMT";

// Database credentials

final String USER = "root";

final String PASS = "1234";

// Set response content type

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String title = "Database Result";

String docType =

"<!doctype html public \"-//w3c//dtd html 4.0 " + "transitional//en\">\n";

out.println(docType +

"<html>\n" +

"<head><title>" + title + "</title></head>\n" +

"<body bgcolor = \"#f0f0f0\">\n" +

"<h1 align = \"center\">" + title + "</h1>\n");

Connection conn = null;

Statement stmt = null;

try {

// Register JDBC driver

Class.forName("com.mysql.cj.jdbc.Driver");

// Open a connection

conn = DriverManager.getConnection(DB\_URL, USER, PASS);

// Execute SQL query

stmt = conn.createStatement();

String sql;

sql = "SELECT id, first, last, age FROM Employees";

ResultSet rs = stmt.executeQuery(sql);

// Extract data from result set

while(rs.next()){

//Retrieve by column name

int id = rs.getInt("id");

int age = rs.getInt("age");

String first = rs.getString("first");

String last = rs.getString("last");

//Display values

out.print("ID: " + id);

out.print(", Age: " + age);

out.print(", First: " + first);

out.print(", Last: " + last);

out.println("<br>");

}

out.println("</body></html>");

// Clean-up environment

rs.close();

stmt.close();

conn.close();

}

catch(SQLException se) {

//Handle errors for JDBC

se.printStackTrace();

}

catch(Exception e) {

//Handle errors for Class.forName

e.printStackTrace();

}

finally {

//finally block used to close resources

try {

if(stmt!=null)

stmt.close();

}

catch(SQLException se2) {

} // nothing we can do

try {

if(conn!=null)

conn.close();

}

catch(SQLException se) {

se.printStackTrace();

} //end finally try

} //end try

}