

[JSP暑假实训] 二.JSP网站创建及Servlet实现注册表单提交、计算器运算

原创 Eastmount 2019-07-01 16:22:31 1834 收藏 2

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法、



Eastmount

¥9.90

订阅

本系列文章是作者暑假给学生进行实训分享的笔记，主要介绍MyEclipse环境下JSP网站开发，包括JAVA基础、网页布局、数据库基础、Servlet、前端后台数据库交互、DAO等知识。

本篇文章开始讲解MyEclipse环境下创建JSP网站，并实现注册页面的创建及Servlet获取提交的数据。非常基础的文章，希望对读者有所帮助，尤其是我的学生。

参考前文：

[Java+MyEclipse+Tomcat \(一\)配置过程及jsp网站开发入门](#)

[Java+MyEclipse+Tomcat \(二\)配置Servlet及简单实现表单提交](#)

[Java+MyEclipse+Tomcat \(三\)配置MySQL及查询数据显示在JSP网页中](#)

[Java+MyEclipse+Tomcat \(四\)Servlet提交表单和数据库操作](#)

[Java+MyEclipse+Tomcat \(五\)DAO和Java Bean实现数据库和界面分开操作](#)

[Java+MyEclipse+Tomcat \(六\)详解Servlet和DAO数据库增删改查操作](#)

前文：

[\[JSP暑假实训\] 一.MyEclipse安装及JAVA基础知识](#)

文章目录

[一.JSP网站创建](#)

[二.JSP注册页面实现](#)

[三.Servlet实现表单提交](#)

[1.什么是Servlet?](#)

[2.创建Servlet](#)

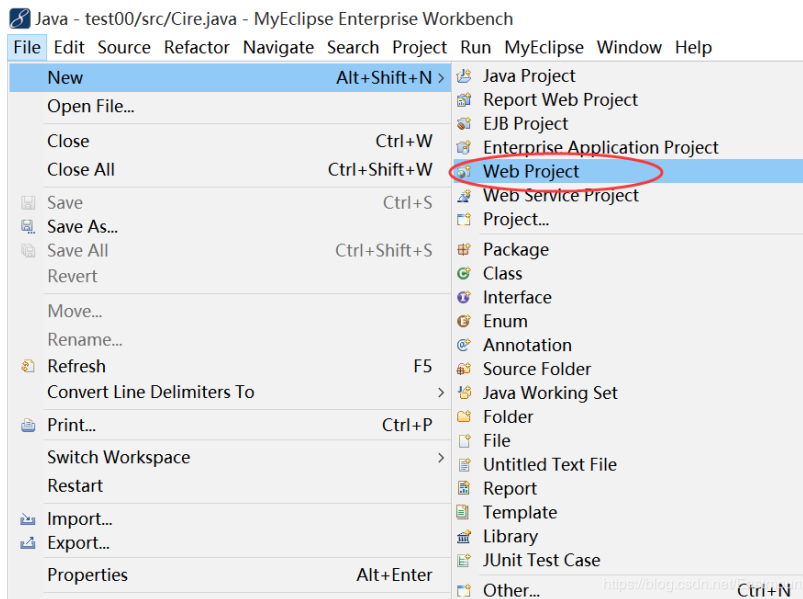
[四.Servlet实现计算器运算](#)

[五.总结](#)

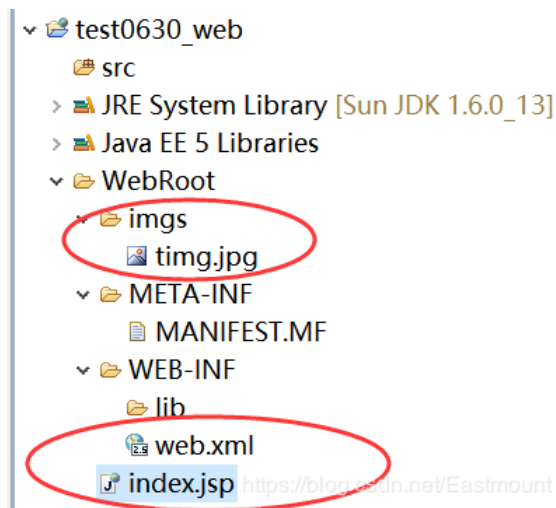
一.JSP网站创建

创建Web Project，工程命名为“test0630_web”。

注意，默认同学们学过HTML，推荐我的专栏：[HTML网站前端设计](#)



创建的工程目录如下图所示：



在WebRoot路径下新建一个文件夹，命名为“imgs”，用于存储图片，然后编辑“index.jsp”主页。

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<%
String path = request.getContextPath();
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<base href="<%=basePath%>">

<title>第一个JSP网站</title>
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
<meta http-equiv="description" content="This is my page">
```

```

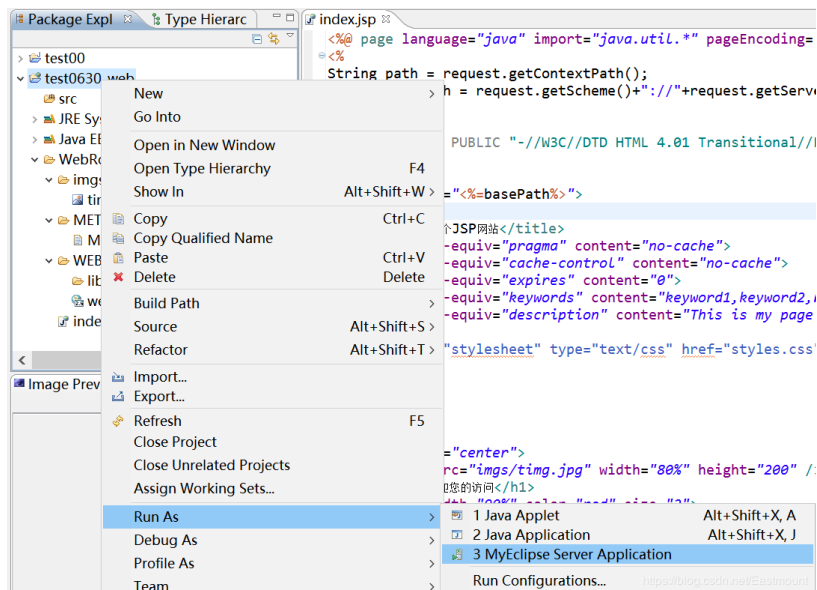
<!--
<link rel="stylesheet" type="text/css" href="styles.css">
-->
</head>

<body>

<div align="center">
    
    <h1>欢迎您的访问</h1>
    <hr width="90%" color="red" size="2">
    贵州纵美路迢迢,<br />
    未付劳心此一遭。<br />
    搜得破书三四本,<br />
    也堪将去教尔曹。<br />
</div>
</body>
</html>

```

右键项目，点击 “Run As” -> “MyEclipse Server Application” 运行网站在本地服务器上。



其运行结果如下图所示：



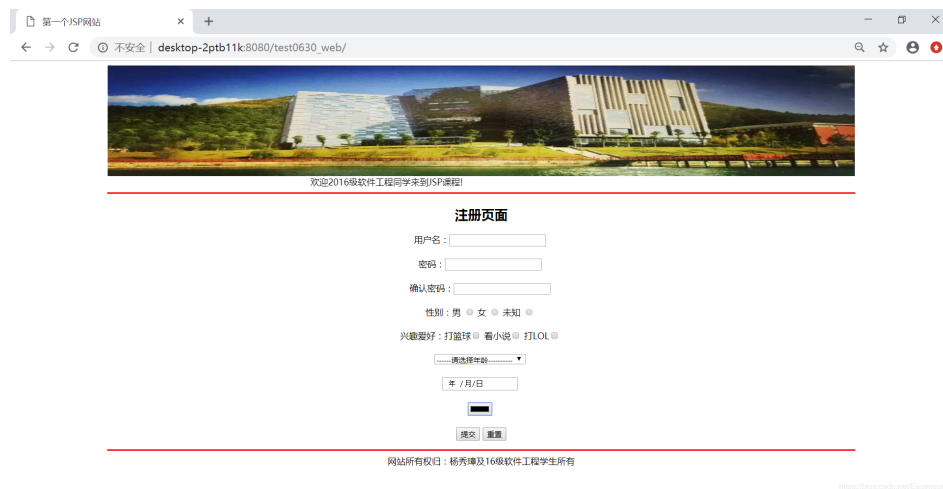
Web文档都有一个唯一的地址，通过URL格式来进行定位，其格式为：

协议://IP地址:端口/站点名/目录/文件名

其中协议主要有HTTP、HTTPS和FTP。根据不同的协议，默认端口可以省略，HTTP/HTTPS为80端口，FTP为21端口。例：<http://210.30.108.30:8080/test/admin/login.jsp>

二.JSP注册页面实现

修改“index.jsp”代码，制作如下图所示的注册页面：



完整代码如下所示：

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<%
String path = request.getContextPath();
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<base href="<%=basePath%>">

<title>第一个JSP网站</title>
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
<meta http-equiv="description" content="This is my page">
<!--
<link rel="stylesheet" type="text/css" href="styles.css">
-->
</head>

<body>

<div align="center">

<marquee scrolldelay="10">欢迎2016级软件工程同学来到JSP课程!</marquee>
<hr width="80%" color="red" size="3" />
<h2>注册页面</h2>
<form method="post" action="/test0630_web/servlet/ZCServlet">
```

```

用户名: <input type="text" name="t1" id="t1" />
<br /><br />

密码: <input type="password" name="t2" id="t2" />
<br /><br />

确认密码: <input type="password" name="t3" id="t3" />
<br /><br />

性别: 男 <input type="radio" value="男" name="sex" id="t4" />
      女 <input type="radio" value="女" name="sex" id="t5" />
      未知 <input type="radio" value="未知" name="sex" id="t6" />

<br /><br />
兴趣爱好: 打篮球<input type="checkbox" value="打篮球" name="interest" id="t7" />
          看小说<input type="checkbox" value="看小说" name="interest" id="t8" />
          打LOL<input type="checkbox" value="打LOL" name="interest" id="t9" />

<br /><br />
<select name="age">
    <option>-----请选择年龄-----</option>
    <option value="00后">00后</option>
    <option value="90后">90后</option>
    <option value="80后">80后</option>
    <option value="70后">70后</option>
</select>

<br /><br />
<input type="date" name="t10" id="t10" />

<br /><br />
<input type="color" name="t11" id="t11" />

<br /><br />

<input type="submit" name="t12" id="t12" />
<input type="reset" name="t13" id="t12" />
</form>
<hr width="80%" color="red" size="3" />
网站所有权归: 杨秀璋及16级软件工程学生所有
</div>
</body>
</html>

```

三.Servlet实现表单提交

1.什么是Servlet?

在Sun公司制定Java EE规范初期, 为实现动态Web而引入了Servlet, 用于替代笨重的CGI(通用网关接口), 实现了Java语言编程的动态Web技术, 奠定了Java EE的基础。后来为进一步简化动态Web网页的生成, 并且在微软公司推出了ASP(Active X服务系统页面)技术的竞争下, Sun推出了JSP规范, 进一步简化了Web网页的编程。但JSP在进行HTTP请求处理方面不如Servlet方便和规范, Servlet在当今的MVC模式Web开发中牢牢占据一地, 并且现在流行的Web框架基本基于Servlet技术, 如Struts、WebWork和Spring MVC等。只有掌握了Servlet才能真正掌握Java Web编程的核心和精髓。

那么, 究竟Servlet是什么呢?

Servlet是运行在Web容器的类, 能处理Web客户的HTTP请求, 并产生HTTP响应。

Servlet是Java EE规范定义的Web组件, 运行在Web容器中。由Web容器负责管理Servlet的声明周期, 包括创建和销毁Servlet对象。客户不能直接创建Servlet对象和调用Servlet的方法, 只能通过向Web服务器发出HTTP请求, 间接调用Servlet的方法。这是Servlet与普通

Java类的重要区别。

Sun在如下两个包中提供了Servlet的全部接口和类：

- javax.servlet包含支持所有协议的通用Web组件接口和类
- javax.servlet.http包含支持HTTP协议的接口和类

Servlet可以完成Web组件具有的所有功能，如下：

- 接收HTTP请求
- 取得请求信息，包括请求头和请求参数数据
- 调用其他Java类方法完成具体的业务功能
- 生成HTTP响应，包括HTML和非HTML响应
- 实现到其他Web组件的跳转，包括重定向和转发

编写Servlet需要引入的两个包和Java I/O包：

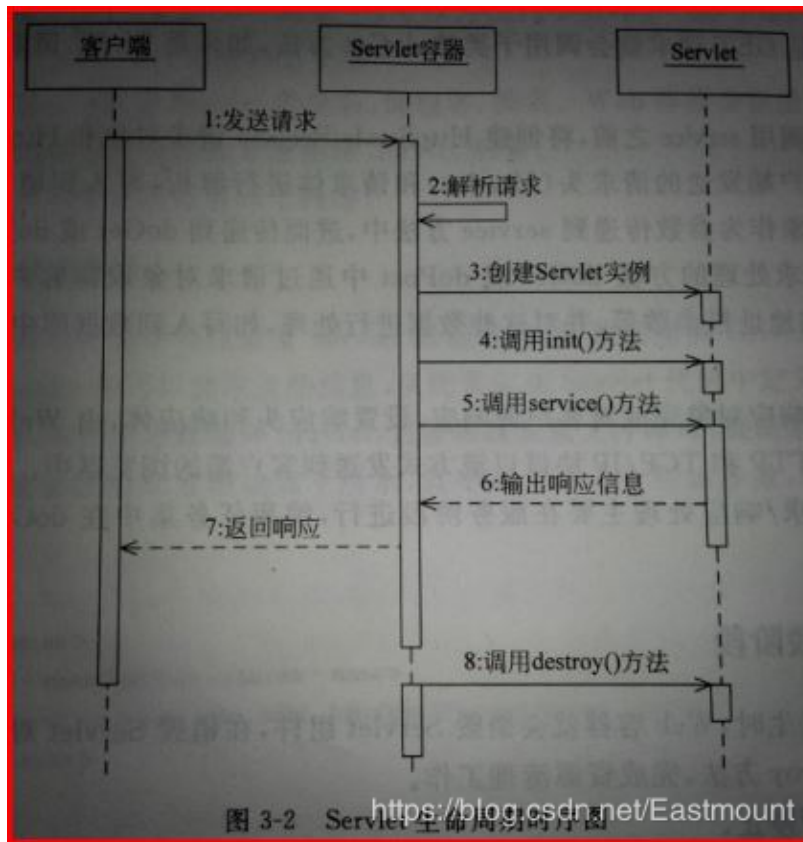
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
```

编写接收HTTP请求并进行HTTP响应的Servlet要继承javax.servlet.http.HttpServlet。Servlet类的定义如下：

```
public class LoginAction extends HttpServlet {}
```

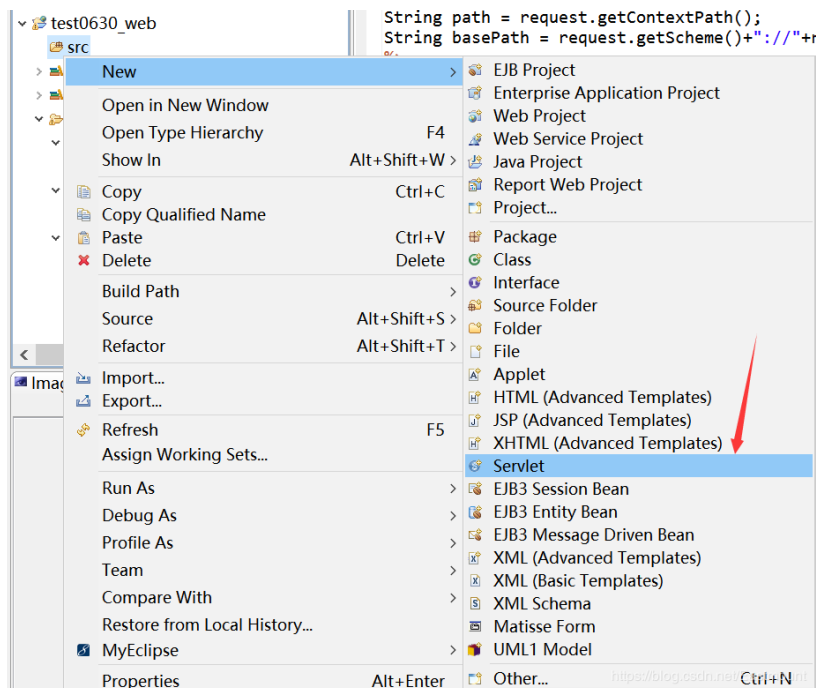
每个Servlet一般都需要重写doGet方法和doPost方法。当用户使用GET方式请求Servlet时，Web容器调用doGet方法处理请求；当用户使用POST方法请求Servlet时，Web容器调用doPost方法。

Servlet是生命周期时序图如上图所示，包括加载类和实例化阶段、初始化阶段、处理请求阶段和销毁阶段。

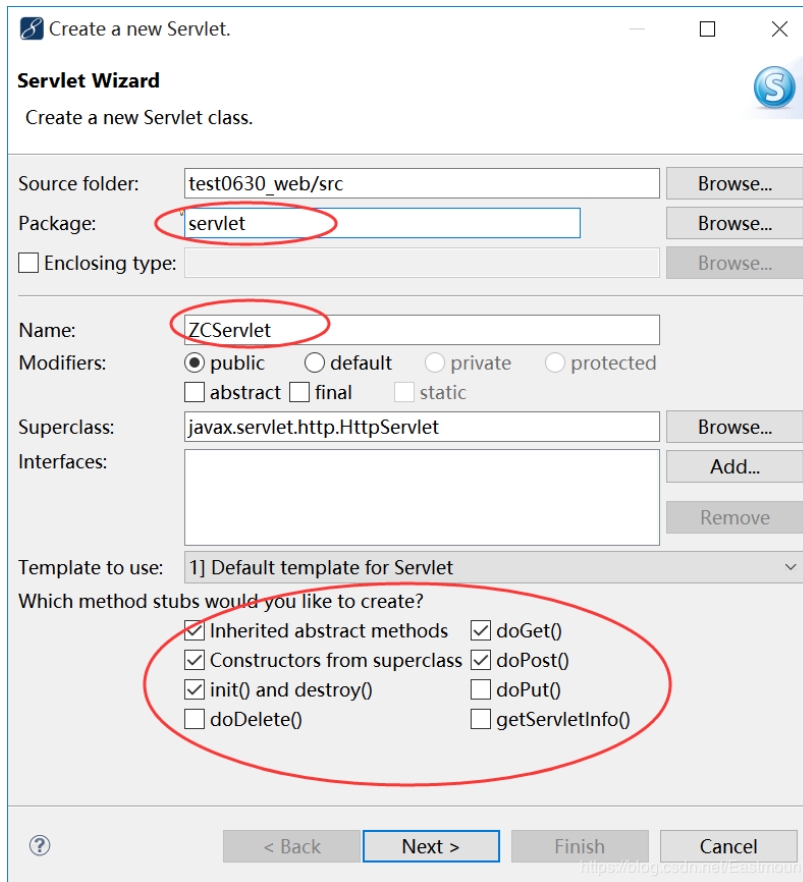


2.创建Servlet

第一步：选中“src”文件夹，然后右键新建一个Servlet文件，部分MyEclipse的Servlet创建位于Other中。



新建Servlet文件名为“ZCServlet”，同时创建在“servlet”文件夹中，如下图所示：



由于Servlet是一个Java类文件，不像JSP那样直接存放在Web目录下就能获得URL请求访问地址。Servlet必须在Web的配置文件/WEB-INF/web.xml中进行配置和映射才能响应HTTP请求。

第二步：新建Servlet之后，会自动生成配置文件，此时的web.xml代码如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <display-name></display-name>

  <servlet>
    <description>This is the description of my J2EE component</description>
    <display-name>This is the display name of my J2EE component</display-name>

    <servlet-name>ZCServlet</servlet-name>
    <servlet-class>servlet.ZCServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>ZCServlet</servlet-name>
    <url-pattern>/servlet/ZCServlet</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```


其中声明Servlet的名字，一般与Servlet的类名相同即可，要求在一个web.xml文件内名字唯一。指定Servlet的全名，即包名.类型。Web容器会根据此定义载入类文件到内容中，进而调用默认构造方法创建Servlet对象。

Servlet映射：任何Web文档在Internet上都要有一个URL地址才能被请求访问，Servlet不能像JSP一样直接放在Web的发布目录下，因此Servlet需要单独映射URL地址。在WEB-INF/web.xml中进行Servlet的URL映射。即：

```
<servlet-mapping>
    <servlet-name>ZCServlet</servlet-name>
    <url-pattern>/servlet/ZCServlet</url-pattern>
</servlet-mapping>
```

上面代码是绝对地址方式映射，只能映射到一个地址。其中Servlet-name名称和Servlet声明中要一致，URL的格式如下：/目录/目录/文件名.扩展名。

第三步：编辑ZCServlet类中的代码，涉及GET方法或POST方法。代码如下所示，主要修改doPost方法。

```
package servlet;

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

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ZCServlet extends HttpServlet {

    /**
     * Constructor of the object.
     */
    public ZCServlet() {
        super();
    }

    /**
     * Destruction of the servlet. <br>
     */
    public void destroy() {
        super.destroy(); // Just puts "destroy" string in log
        // Put your code here
    }

    /**
     * The doGet method of the servlet. <br>
     *
     * This method is called when a form has its tag value method equals to get.
     *
     * @param request the request send by the client to the server
     * @param response the response send by the server to the client
     * @throws ServletException if an error occurred
     * @throws IOException if an error occurred
     */
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\">");
        out.println("<HTML>");
        out.println("    <HEAD><TITLE>A Servlet</TITLE></HEAD>");
```

```

        out.println(" <BODY>");
        out.print("    This is ");
        out.print(this.getClass());
        out.println(", using the GET method");
        out.println(" </BODY>");
        out.println("</HTML>");
        out.flush();
        out.close();
    }

    /**
     * The doPost method of the servlet. <br>
     *
     * This method is called when a form has its tag value method equals to post.
     *
     * @param request the request send by the client to the server
     * @param response the response send by the server to the client
     * @throws ServletException if an error occurred
     * @throws IOException if an error occurred
     */
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        //设置编码方式 防止乱码
        request.setCharacterEncoding("UTF-8");
        response.setCharacterEncoding("UTF-8");

        //获取用户名
        String name = request.getParameter("t1");
        //密码
        String pwd = request.getParameter("t2");
        //性别
        String sex = request.getParameter("sex");
        //兴趣
        String [] interest = request.getParameterValues("interest");
        //年龄
        String age = request.getParameter("age");
        //时间
        String time = request.getParameter("t10");
        //颜色
        String color = request.getParameter("t11");

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out

            .println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\">");
        out.println("<HTML>");
        out.println("    <HEAD><TITLE>A Servlet</TITLE></HEAD>");
        out.println("    <BODY>");
        out.print("    This is ");
        out.print(this.getClass());
        out.println(", using the POST method");

        //输出获取的信息
        out.println("<h2>我注册的用户名为: " + name + "</h2>");
        out.println("<h2>我注册的用户密码为: " + pwd + "</h2>");
        out.println("<h2>我注册的用户性别为: " + sex + "</h2>");
        out.println("<H2>兴趣");
        for(String str:interest) {
            out.println(str+" ");
        }
        out.println("<h2>我注册的年龄为: " + age + "</h2>");
    }

```

```

        out.println("<h2>我注册的时间为: " + time + "</h2>");
        out.println("<h2>我喜欢的颜色为: " + color + "</h2>");

        out.println("  </BODY>");
        out.println("</HTML>");
        out.flush();
        out.close();
    }

    /**
     * Initialization of the servlet. <br>
     *
     * @throws ServletException if an error occurs
     */
    public void init() throws ServletException {
        // Put your code here
    }
}

```

第四步：修改JSP中表单提交的内容，当用户注册点击提交时自动跳转至结果页面。

```

<form method="post" action="/test0630_web/servlet/ZCServlet">

```

第五步：运行程序及反馈结果。

当用户输出信息并点击提交按钮之后，如下所示：

欢迎2016级软件工程同学来到JSP课程!

注册页面

用户名:

密码:

确认密码:

性别: 男 ☒ 女 ☐ 未知 ☐

兴趣爱好: 打篮球 ☒ 看小说 ☒ 打LOL ☐

00后

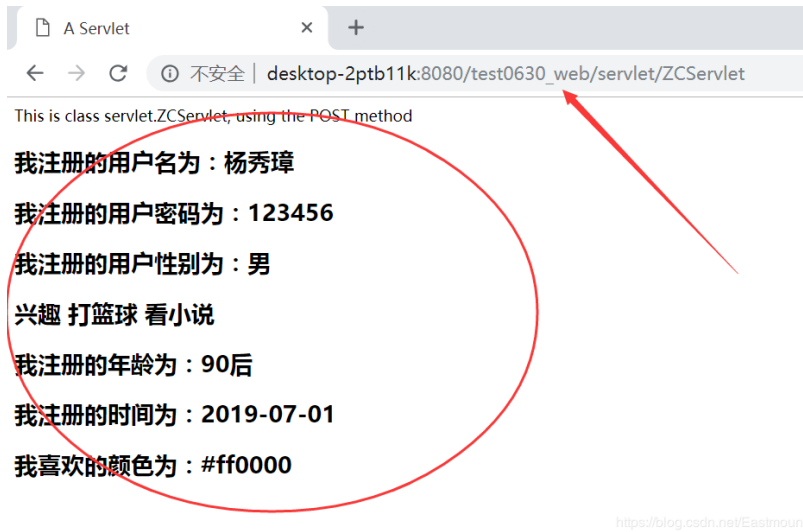
2019/07/01

网站所有权归：杨秀璋及16级软件工程学生所有

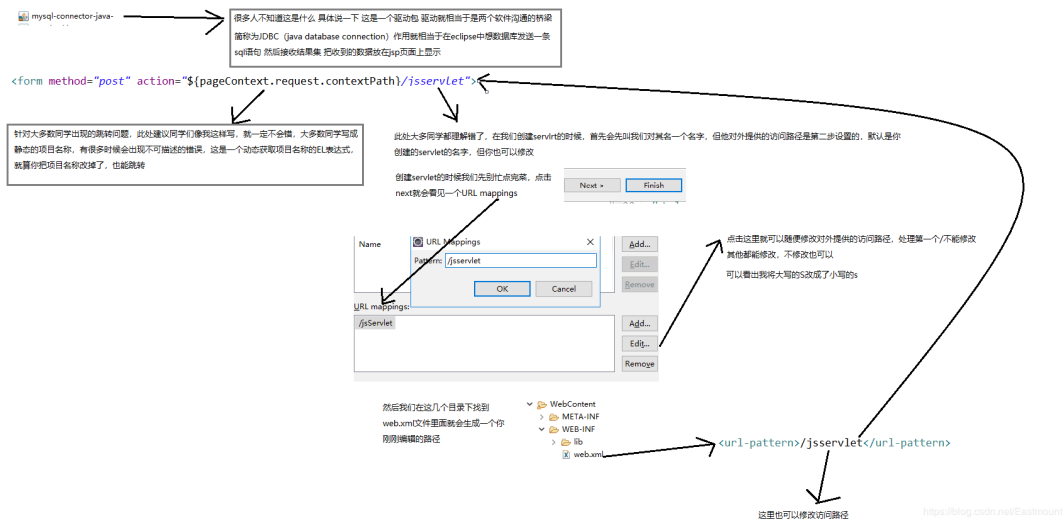
<https://blog.csdn.net/Eastmount>

“杨秀璋”同学的注册信息如下所示，其网址为：

http://desktop-2ptb11k:8080/test0630_web/servlet/ZCServlet



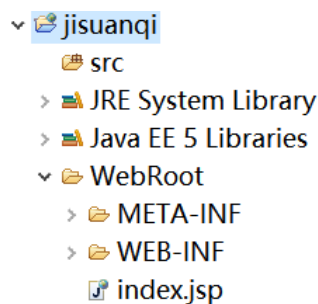
注意：学生“王志能”补充了如何修改访问路径的问题，感谢！同时，编码方式统一设置为UTF-8。



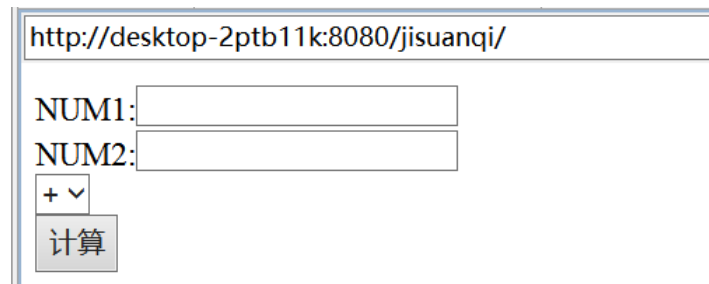
四.Servlet实现计算器运算

下面补充一个案例，完整的介绍Servlet实现计算机的运算。

第一步：新建一个Web项目，项目名称为“jisuanqi”。



第二步：书写表单，在表单中可以输入需要计算的两个数，选择计算符号，如下图所示。



对应代码如下：

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<%
String path = request.getContextPath();
String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";
%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <base href="<%=basePath%>">

    <title>My JSP 'index.jsp' starting page</title>
    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta http-equiv="keywords" content="keyword1,keyword2,keyword3">
    <meta http-equiv="description" content="This is my page">
    <!--
    <link rel="stylesheet" type="text/css" href="styles.css">
    -->
  </head>

  <body>

    <form action="servletCal" method="post">
      NUM1:<input type="text" name="num1"/><br/>
      NUM2:<input type="text" name="num2"/><br/>
      <select name="pro">
        <option>+</option>
        <option>-</option>
        <option>*</option>
        <option>/</option>
      </select><br/>
      <input type="submit" value="计算">
    </form>

  </body>
</html>
```

第三步：前端表单书写完毕之后便可以书写后台代码了，后台主要负责前端数据的获取和计算。后台主要分为两个类Cal.java、servletCal.java。其中Cal.java主要负责计算：

```
package servlet;

public class Cal {
```

```

private int num1;    //数字
private int num2;    //数字
private String pro;  //运算

public void setNum1(int num1) {
    this.num1 = num1;
}
public void setNum2(int num2) {
    this.num2 = num2;
}
double sum=0;
//加法
public double add()
{
    sum=num1+num2;
    return sum;
}
//减法
public double sub()
{
    sum=num1-num2;
    return sum;
}
//乘法
public double mult()
{
    sum=num1*num2;
    return sum;
}
//除法
public double div()
{
    sum=num1/num2;
    return sum;
}
}

```

第四步：servletCal.java负责从前台接收数据，数据接收完成后将其传递到算法中计算，将计算结果返回显示在页面上。

```

package servlet;

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

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class servletCal extends HttpServlet {

    /**
     * Constructor of the object.
     */
    public servletCal() {
        super();
    }

    /**
     * Destruction of the servlet. <br>
     */
}

```

```

public void destroy() {
    super.destroy(); // Just puts "destroy" string in log
    // Put your code here
}

/**
 * The doGet method of the servlet. <br>
 *
 * This method is called when a form has its tag value method equals to get.
 *
 * @param request the request send by the client to the server
 * @param response the response send by the server to the client
 * @throws ServletException if an error occurred
 * @throws IOException if an error occurred
 */
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<!DOCTYPE HTML PUBLIC \\"-//W3C//DTD HTML 4.01 Transitional//EN\\>");
    out.println("<HTML>");
    out.println("  <HEAD><TITLE>A Servlet</TITLE></HEAD>");
    out.println("  <BODY>");
    out.print("    This is ");
    out.print(this.getClass());
    out.println(", using the GET method");
    out.println("  </BODY>");
    out.println("</HTML>");
    out.flush();
    out.close();
}

/**
 * The doPost method of the servlet. <br>
 *
 * This method is called when a form has its tag value method equals to post.
 *
 * @param request the request send by the client to the server
 * @param response the response send by the server to the client
 * @throws ServletException if an error occurred
 * @throws IOException if an error occurred
 */
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    // 获取数据并计算
    String num1=request.getParameter("num1");
    String num2=request.getParameter("num2");
    String pro=request.getParameter("pro");

    Cal cal=new Cal();
    cal.setNum1(Integer.parseInt(num1));
    cal.setNum2(Integer.parseInt(num2));

    double result=0;
    if(pro.equals("+"))
    {
        result=cal.add();
    }else if(pro.equals("-"))
    {
        result=cal.sub();
    }
}

```

```

    }else if(pro.equals("*"))
    {
        result=cal.mult();
    }else if(pro.equals("/"))
    {
        result=cal.div();
    }

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\">");
    out.println("<HTML>");
    out.println("  <HEAD><TITLE>A Servlet</TITLE></HEAD>");
    out.println("  <BODY>");
    out.println("    <h1>The Result is " + result + "</h1>");
    out.println("  </BODY>");
    out.println("</HTML>");
    out.flush();
    out.close();
}

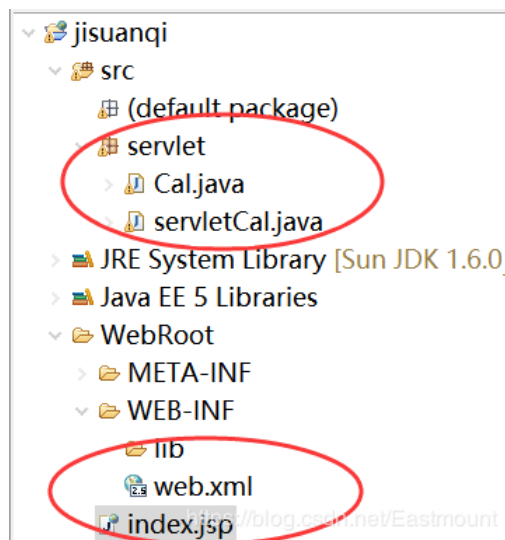
/**
 * Initialization of the servlet. <br>
 *
 * @throws ServletException if an error occurs
 */
public void init() throws ServletException {
    // Put your code here
}
}

```

第五步：修改index.jsp文件中的表单form提交代码，即action参数。

```
<form action="/jisuanqi/servlet/servletCal" method="post">
```

此时程序的目录如下所示：



第六步：运行程序并显示相关结果。

http://desktop-2ptb11k:8080/jisuanqi/

NUM1: 12

NUM2: 4

* ▾

计算

<https://blog.csdn.net/Eastmount>

http://desktop-2ptb11k:8080/jisuanqi/servlet/servletCal

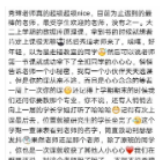
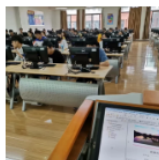
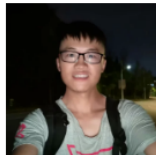
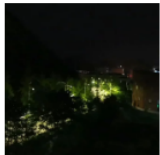
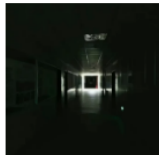
The Result is 48.0

五.总结

最近连续十多天给学生们分享暑假实训，JSP网站开发，周末不间断。大周末深夜凌晨办公室备课，为了什么？回想三年来，挺感谢学生们的厚爱的，无以回报，只希望自己能好好分享知识，认真教好每一位学生，将心比心，也望自己博士学成归来，能继续教书育人，感恩家乡和女神。

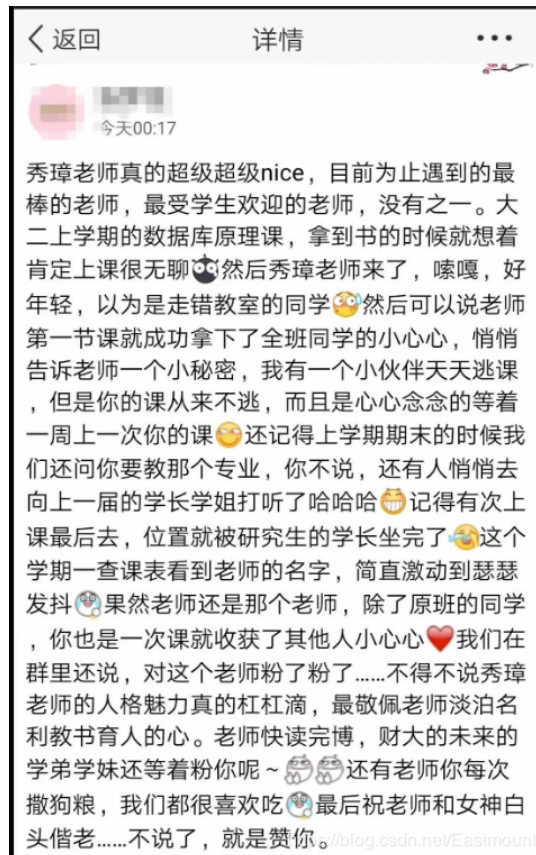


Eastmountain 凌晨备课走出办公室，有点惨得慌，感觉整个校园路上就我一个人，应该没有鬼吧！还好女神打来电话，自拍完赶紧跑，“鬼来了，啊”😱
明早继续上课，加油。真的很感谢同学们的厚爱，无以回报，只能好好分享知识，认真教好每一位学生，将心比心，晚安娜🌙



昨天0:40

<https://blog.csdn.net/Eastmount> 评论(6) 转发 更多 ▾



(By:Eastmount 2019-07-01 中午1点 <http://blog.csdn.net/eastmount/>)



Eastmount 博客专家

原创文章 459 获赞 6629 访问量 517万+

关注

他的留言板