第 1 章

初试 Servlet/JSP 编程

Chapter 1

1.1 Servlet 程序

知识要点

1. Servlet 的基本概念
2. Tomcat 的安装配置

网页本身是一种文本文件，而 HTML(HyperText Marked Language)则是编写网页的主要语言。

例 1-1 利用 HTML 编写一个输出 Hello World 的网页。

```
<html>
  <head>
    <title>一个简单的 HTML 示例</title>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```

将以上源代码保存在文件 helloworld.html 中，用浏览器打开，可以看到图 1-1 所示的效果。

图 1-1 HTML 示例

HelloWorld-Servlet.wmv

Servlet 是用 Java 程序编写的服务器端程序，这一部分讨论了如何用 Java 编写 Servlet，以及如何实现服务器端与客户端的动态交互。

1. 实例

用 Java 编写 Servlet 程序，先发送一个 HTML 网页。

2. 分析

编写 Servelt 程序，首先应该继承 HttpServlet，然后根据是 GET 的 POST 请求，覆盖

doGet()、doPost()方法。

3. 代码实现

```
1.  import java.io.*;
2.  import javax.servlet.*;
3.  import javax.servlet.http.*;

4.  public class HelloWorld extends HttpServlet {
5.    public void doGet(HttpServletRequest request, HttpServletResponse response)
6.    throws ServletException, IOException {
7.      response.setContentType("text/html");
8.      PrintWriter out=response.getWriter();
9.      out.println("<html>");
10.     out.println("<head><title>Hello World</title></head>");
11.     out.println("<body>");
12.     out.println("<h1>Hello World</h1>");
13.     out.println("</body></html>");
14.   }
15.   public void doPost(HttpServletRequest request,
16.   HttpServletResponse response)
17.   throws ServletException, IOException {
18.     doGet(request, response);
19.   }
20. }
```

程序中第7～13行向客户端浏览器发送网页。out是PrintWriter对象，println()方法把网页作为字符串发送到客户端浏览器。

Tomcat 安装和运行.wmv

4. 测试与运行

程序编写好后，还需选择一个Web容器发布Servlet动态网页，本书用的是Tomcat。下面介绍如何配置Tomcat环境，以及如何运行该程序。

(1) Tomcat的安装

支持Servlet/JSP的容器有很多，Tomcat是SUN公司官方推荐的Servlet和JSP容器。

在Windows环境下，运行Tomcat需设置TOMCAT_HOME和JAVA_HOME两个环境变量，假设Tomcat安装在D:\tomcat8目录下，环境变量设置如图1-2所示。

假设JDK安装在D:\Program Files\Java\jdk1.8.0_131目录下，JAVA_HOME设置如图1-3所示。

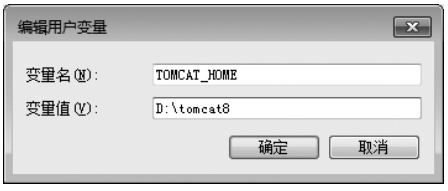

图 1-2　设置 TOMCAT 环境变量

图 1-3　设置 Java 环境变量

(2) 启动和测试 Tomcat

设置完毕就可以运行 Tomcat 服务器了。进入 Tomcat 安装目录下的 bin 目录,运行 startup 程序,启动 Tomcat,运行 shutdown 程序,关闭 Tomcat。启动完成后,在浏览器中输入 http://localhost:8080/,如图 1-4 所示。

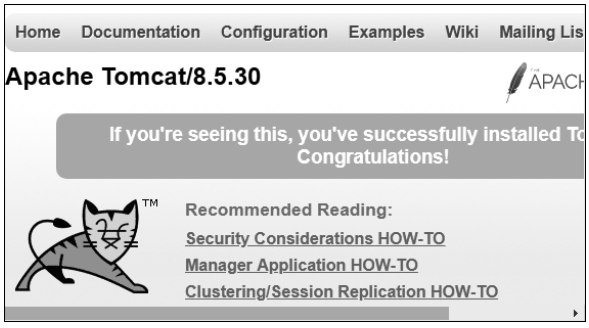

图 1-4 正确启动 Tomcat 后的界面

(3) 编译

在编译 Servlet 程序之前,要确保 servlet.jar 和 jsp.jar 加入 CLASSPATH 变量。CLASSPATH 变量应该包含当前目录(即在 CLASSPATH 中包含".")。

Tomcat 安装目录下的 common\lib 目录有一个文件 servlet-api.jar,把它加入到 CLASSPATH 变量中也可以。如:set classpath=%classpath%;C:\tomcat\common\lib\servlet-api.jar。

或者在编译的时候给出 classpath 的路径,如:javac HelloWorld.java-classpath C:\tomcat\common\lib\servlet-api.jar。HelloWorld.java 是要编译的文件名。

Tomcat 中的应用程序按一定目录结构来组织,Web 应用下的 WEB-INF 目录很重要,通常在 WEB-INF 目录下有一个 web.xml 文件和一个 classes 目录,web.xml 是该应用的配置文件,而 classes 目录下则包含编译好的 Servlet 类以及 JSP 或 Servlet 所依赖的其他类。应用程序所依赖的类可以打包成 JAR 文件放到 WEB-INF 下的 lib 目录下,虽然也可以放到系统的 CLASSPATH 变量中,但那样做会导致程序的移植和管理困难。编译后的类文件 HelloWorld.class 需放置在<CATALINA_HOME>/webapps/ROOT/WEB-INF/classes 目录下。

(4) 部署 Servlet

Servlet 是用 Java 编写的一个服务器端程序,还需要为它配置一个 URL 地址。为此需要在文件 web.xml 中配置。在文件中加入下面的内容。

```
<servlet>
    <servlet-name>HelloWorld</servlet-name>
    <servlet-class>HelloWorld</servlet-class>
</servlet>

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

在配置文件中，＜servlet＞标签把 HelloWorld 类配置为名为 HelloWorld 的 Servlet，＜servlet-mapping＞标签为该 Servlet 配置 URL 地址：/servlet/HelloWorld。

注意 较高版本的 JDK 可直接在程序中用注释语句把 HelloWorld 类配置为一个 Servlet，语句如下：

```
@WebServlet("/servlet/HelloWorld")
public class HelloWorld extends HttpServlet {...}
```

第一行@WebServlet 是注释语句，把 HelloWorld 类配置成 URL 地址为/servlet/HelloWorld 的一个 Servlet。

(5) Servlet 程序的运行

启动 Tomcat，在地址栏中输入 http://localhost:8080/servlet/HelloWorld，可以看到程序的运行结果如图 1-5 所示。

图 1-5 运行 Servlet 程序的结果

5. 技术分析

(1) CGI 和 Servlet 概念

用户一般在用户界面通过一个表单填写数据并传送到 Server，让 Server 的 CGI 程序处理。CGI(Common Gateway Interface)是一个用于 Web 服务器与外部程序之间交互的标准接口，使得外部程序能生成 HTML、图像或者其他内容，而服务器处理的方式与那些非外部程序生成的 HTML、图像或其他内容的处理方式是相同的。因此，CGI 程序不仅能生成静态内容，而且也能生成动态内容。CGI 是一个定义良好并被广泛支持的标准。

Servlet 相当于使用 Java 技术的 CGI。Servlet 程序在服务器端运行，动态地生成 Web 页面。与传统的 CGI 和许多其他类似 CGI 的技术相比，Java Servlet 具有更高的效率，其功能更强大，具有更好的可移植性，更节省资金。

(2) Servlet 处理请求方式

Servlet 可以处理 GET 请求。所谓的 GET 请求，可以把它看成是用户提交没有指定 METHOD 的表单所发出的请求。Servlet 也可以很方便地处理 POST 请求，POST 请求是用户提交那些指定了 method＝"POST"的表单时所发出的请求。

Servlet 的 doGet()和 doPost()方法都有两个参数，分别为 HttpServletRequest 类型和 HttpServletResponse 类型。HttpServletRequest 类型提供访问有关请求的信息，例如表单数据、HTTP 请求头等。HttpServletResponse 除了提供用于指定 HTTP 应答状态(200、404 等)、应答头(Content-Type、Set-Cookie 等)的方法之外，最重要的是它提供了一个用于向客户端发送数据的 PrintWriter。对于简单的 Servlet 来说，它的大部分工作是通过 println 语句生成向客户端发送的页面。

doGet()和 doPost()方法会抛出两个异常，因此必须在声明中包含二者。另外，还必须导入 java.io 包(包含 PrintWriter 等类)、javax.Servlet 包(包含 HttpServlet 等类)以及 javax.

Servlet.http 包(包含 HttpServletRequest 类和 HttpServletResponse 类)。

doGet()和 doPost()这两个方法是由 service()方法调用的,有时可能需要直接覆盖 service()方法。Servlet 处理 GET 和 POST 两种请求时,要输出 HTML 还有两个额外的步骤要做:告诉浏览器接下来发送的是 HTML;修改 println()语句,构造出合法的 HTML 页面。

6. 问题与思考

试编写 Servlet 程序,输出自己的姓名。

提示 为了防止乱码,最好在 Servlet 程序中加上以下两条语句。

```
response.setContentType("text/html;charset=GBK");
request.setCharacterEncoding("GBK");
```

1.2 JSP 程 序

知识要点

1. JSP 的基本概念
2. JSP 在 Tomcat 中的配置

HelloWorld-JSP.wmv

JSP(Java Server Pages)是一种实现普通静态 HTML 和动态 HTML 混合编码的技术,许多由 CGI 程序生成的页面大部分仍旧是静态 HTML,动态内容只在页面中有限的几个部分出现。但是包括 Servlet 在内的大多数 CGI 技术及其变种总是通过程序生成整个页面。

1. 实例

用 Java 编写 JSP 程序,发送一个 HTML 网页。

2. 分析

JSP 遵循脚本语言的编制标准,是为解决 Servlet 中编程的困难而开发的技术,因此 JSP 在程序的编写方面比 Servlet 要容易得多。

3. 代码实现

```
1.  <html>
2.    <head>
3.      <title>一个简单的 JSP 示例</title>
4.    </head>
5.    <body>
6.      <h1><%out.println("Hello World");%></h1>
7.    </body>
8.  <html>
```

程序中的第 6 行向浏览器发送 Hello World。程序中 out 是 JSP 的默认对象,可以直接引用,不像在 Servlet 中需先定义。该语句可以简写为<%="Hello World"%>。

4. 测试与运行

继续用 Tomcat 作为 Web 容器。把该文件存放在＜CATALINA_HOME＞/webapps/ROOT 目录下，然后在浏览器地址栏中输入 http://localhost:8080/HelloWorld.jsp，JSP 程序运行效果如图 1-6 所示。

图 1-6　运行 JSP 程序的结果

5. 技术分析

Servlet 是一种在服务器端运行的 Java 程序，Sun 首先发展 Servlet，其功能较强，体系设计也较先进，只是它输出 HTML 语句还是采用了老的 CGI 方式，是一句一句输出，所以编写和修改 HTML 非常不方便。

后来 Sun 推出 JSP，把 JSP 标签(Tag)镶嵌到 HTML 语句中，大大简化和方便了网页的设计与修改。虽然 JSP 是以 Servlet 为基础开发的，但两者有诸多不同，Servlet 与 JSP 区别如下。

(1) 编程方式不同

JSP 是为了解决 Servlet 中相对困难的编程技术而开发，因此，JSP 在程序的编写方面比 Servlet 要容易得多，Servlet 严格遵循 Java 语言的编程标准，而 JSP 则遵循脚本语言的编程标准。

(2) Servlet 必须在编译以后才能执行

JSP 并不需要另外进行编译，JSP 容器会自动完成这项工作，而 Servlet 在每次修改代码之后都需要编译完才能执行。

(3) 运行速度不同

由于 JSP 容器将 JSP 程序编译成 Servlet 的时候需要一些时间，所以 JSP 的运行速度比 Servlet 要慢一些。不过 JSP 页面处理过程能够做到一次加载可多次重复执行，即 JSP 容器接到请求以后会确认传递过来的 JSP 是否有改动，如果没有改动，将直接调用 JSP 编译过的 Servlet 类，并提供给客户端解释执行；如果 JSP 文件有所改变，JSP 容器将重新将它编译成 Servlet 类，然后再提交给客户端。

6. 问题与思考

编写 JSP 程序，输出自己的姓名。

提示　为了防止乱码，最好在 JSP 中第一行加上语句：

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

第 2 章

HTML 入 门

Chapter 2

2.1 HTML 基本知识

知识要点

1. HTML 的基本结构
2. HTML 的＜a＞标签
3. HTML 的＜table＞标签

用 HTML 编写的超文本文档称为 HTML 文档，它独立于操作系统平台。一直被用作 WWW（World Wide Web）的信息表示语言，用于 Homepage 的格式设计和 WWW 上 Homepage 之间的连接信息。使用 HTML 语言描述的文件，需要通过 WWW 浏览器显示出效果。

1. 实例

下面通过实例介绍一个很重要的实现超链接的＜a＞标签，设计一个链接到例 1-1 的网页。

2. 分析

锚标签＜a＞＜/a＞用来定义一个超链接，超链接的目标用 href 属性来定义。链接可以是绝对的，比如完整的 http://www.baidu.com；也可以是相对的，即相对于当前页面来说。所以，假设有另外一个 HTML 文档 hello.html，而该文档又恰与当前页面在同一个目录下，其中一行代码可以写成＜a href＝"hello.html"＞到另一个网页＜/a＞。

超链接不仅可以链接到其他的 HTML 文档，也可以链接到页面上的其他文件。

3. 代码实现

```
1.  <html>
2.    <head>
3.      <title>超链接</title>
4.    </head>
5.    <body>
6.      单击<a href="helloworld.html">这里</a>转到其他网页
7.    </body>
8.  </html>
```

程序中第 6 行＜a＞标签的 href 参数列出链接的网页。

4. 测试与运行

在浏览器中打开 link.html 后，效果如图 2-1 所示。

图 2-1　打开该网页后的效果

单击“这里”超链接，转到了另一个网页，如图 2-2 所示。

图 2-2　链接的网页

5. 技术分析

HTML 可以加入图片、声音、动画、影视等内容，还可以从一个文件跳转到另一个文件，与世界各地主机的文件相链接。

(1) HTML 的基本结构

超文本文档分文档头和文档体两部分。在文档头里对这个文档进行了一些必要的定义，而在文档体中才是要显示的各种文档信息。

```
<html>
    <head>
        头部信息
    </head>
    <body>
        文档主体,正文部分
    </body>
</html>
```

HTML 语言使用标签对的方法编写文件，既简单又方便，它通常使用＜标签名＞＜/标签名＞来表示标签的开始和结束(例如＜html＞＜/html＞标签对)，因此，在 HTML 文档中这样的标签对都必须是成对使用的。下面首先介绍一下 HTML 的基本标签。

(2) ＜html＞＜/html＞

＜html＞标签用于 HTML 文档的最前边，用来标识 HTML 文档的开始。而＜/html＞标签则恰恰相反，它放在 HTML 文档的最后边，用来标识 HTML 文档的结束。两个标签必须成对使用。

(3) <head></head>

<head>和</head>构成 HTML 文档的开头部分,在此标签对之间可以使用<title></title>、<script></script>等标签对,这些标签对都是用来描述 HTML 文档相关信息的,<head></head>标签对之间的内容是不会显示出来的,所有标签必须成对使用。

(4) <body></body>

<body></body>是 HTML 文档的主体部分,在此标签对之间可包含 、 、<h1>、</h1>、
、<hr>等众多的标签,它们所定义的文本、图像等将会显示出来。其中<html>在最外层,表示这对标签间的内容是 HTML 文档。有一些 Homepage 省略<html>标签,因为.html 或.htm 文件被 Web 浏览器默认为是 HTML 文档。<head> 之间包括文档的头部信息,如文档总标题等,若不需头部信息,则可省略此标签。<body> 标签一般不省略,表示正文内容的开始。

(5) <title></title>

使用过浏览器的人可能都会注意到浏览器窗口最上边蓝色部分显示的文本信息,那些信息一般是网页的"主题",要将网页的主题显示到浏览器的顶部其实很简单,只要在<title></title>标签对之间加入要显示的文本即可。

注意 <title></title>标签对只能放在<head></head>标签对之间。

(6) 表格

表格由<table>标签来定义。每个表格均有若干行,由<tr>标签定义,每行被分割为若干单元格,由<td>标签定义。字母 td 是指表格数据(table data),即数据单元格的内容。数据单元格可以包含文本、图片、列表、段落、表单、水平线、表格等。

```
<table border="1">
    <tr>
        <td>row 1, cell 1</td>
        <td>row 1, cell 2</td>
    </tr>
    <tr>
        <td>row 2, cell 1</td>
        <td>row 2, cell 2</td>
    </tr>
</table>
```

浏览器显示结果如图 2-3 所示。

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

图 2-3 简单表格效果

如果不定义边框属性,表格将不显示边框。

6. 问题与思考

(1) 利用超链接标签<a>编写网页,当单击本校名称后,链接到本校的网站。

(2) 分别用 Servlet 和 JSP 编写程序,当一个表单提交后,提示接收到了表单请求。

(3) 用 JSP 编写程序,输出九九乘法表。

2.2 在 HTML 中运用 CSS

知识要点

1. CSS 样式规则构成
2. CSS 的选择器
3. CSS 操作字体
4. CSS 控制文字定位
5. CSS 改变列表前标识
6. CSS 控制元素的背景
7. CSS 控制空格及边框

CSS(Cascading Style Sheet)即样式表。CSS 可以扩展 HTML 的功能。

1. 实例

编写程序 HTML 网页,设置超链接和<p>标签的字体和背景颜色。

2. 分析

CSS 可以重新定义 HTML 元素的显示方式。CSS 样式表通常在 HTML 里的<style>标签中定义。

3. 代码实现

```
1.   <html>
2.     <head>
3.       <style>
4.         a {color: red}
5.         p {background-color:blue; color:white}
6.       </style>
7.     </head>
8.     <body>
9.         <a href="http://www.sziit.com.cn">深圳信息职业技术学院</a>
10.        <p>注意字体和背景颜色</p>
11.    </body>
12.  </html>
```

程序第 3～6 行中,a {color: red}用于设置超链接文字颜色为红色,p {background-color: blue; color:white}用于设置<p>标签的字体颜色为白色,背景用蓝色。

4. 测试与运行

用浏览器打开 cssshow.html 网页,结果如图 2-4 所示。

从结果中可以看到,超链接文字是红色字体,在<p>标签中的内容是白色字体、蓝色背景,这都是由网页中的<style>标签设置的效果。

图 2-4 CSS 示例