

第一篇

JSP 与 Servlet

- 第 1 章 JSP 技术简介
- 第 2 章 搭建开发环境
- 第 3 章 JSP 语法详解
- 第 4 章 JSP 内置对象详解
- 第 5 章 在 JSP 中使用 JavaBean
- 第 6 章 Servlet 技术应用
- 第 7 章 过滤器
- 第 8 章 监听器

JSP 技术简介

随着 Internet 技术的迅猛发展与 Java 语言的不断完善优化,越来越多的 Web 程序员义无反顾地踏上了 Java Web 应用开发之路。那么什么是 JSP? 什么是 Servlet? 它们与 Java 语言有什么关系呢? 下面我们就开始 JSP 技术的学习之路。

本章要点:

- 常用的动态网页介绍
- Java、JSP、Servlet 的联系
- JSP 工作原理
- 为什么使用 JSP
- 开发第一个 JSP 页面

1.1 什么是 JSP

JSP 全称是 Java Server Pages,是由 Sun 公司倡导、许多公司参与,于 1999 年推出的一种动态网页技术标准。在传统的网页 HTML 文件 (*.htm, *.html)中加入 Java 程序片段(Scriptlet)和 JSP 标记(Tag),就构成了 JSP 网页 (*.jsp)。网页一般又称 HTML 文件,是一种可以在 WWW 上传输、能被浏览器认识和翻译成页面并显示出来的文件。目前网页根据生成方式,大致可以分为静态网页和动态网页两种。JSP 是基于 Java Servlet 以及整个 Java 体系的 Web 开发技术,利用这一技术可以建立安全、跨平台的先进动态网站。JSP 以 Java 技术为基础,又在许多方面做了改进,具有动态页面与静态页面分离、能够脱离硬件平台的束缚,以及编译后运行等优点。到目前为止,JSP 技术已经逐渐成为 Internet 上的主流开发技术。

需要强调的一点是:要想真正地掌握 JSP 技术,必须有较好的 Java 语言基础,以及 HTML 语言方面的知识。

下面是一个比较简单的 JSP 网页。

```
<%@ page contentType="text/html;charset=GB2312"%>
<%@ page import="java.util. *" %>
<HTML>
<BODY>
<P> 现在的时间是:
    <% Date date=new Date(); %>
<BR>
```

```
<% = date%>  
</BODY>  
</HTML>
```

1.2 动态网页技术

这里说的动态网页,与网页上的各种动画、滚动字幕等视觉上的“动态效果”没有直接关系。动态网页可以是纯文字内容的,也可以是包含各种动画的内容,这些只是网页具体内容的表现形式。无论网页是否具有动态效果,采用动态网站技术生成的网页都称为动态网页。动态网页技术有以下几个特点。

(1) 交互性。即网页会根据用户的要求和选择而动态改变和响应,将浏览器作为客户端界面,这将是今后 WEB 发展的大势所趋。

(2) 自动更新。即无须手动更新 HTML 文档,便会自动生成新的页面,可以大大节省工作量。

(3) 因时因人而变。即当不同的时间、不同的人访问同一网址时会产生不同的页面。

目前,最常用的三种动态网页技术有 ASP(Active Server Pages)、JSP(Java Server Pages)、PHP(PHP: Hypertext Preprocessor)。下面,我们分别从技术和商业的角度来简单介绍这些动态网页技术。

1. ASP

(1) 技术角度

① 简介: ASP 又被称为服务器端的 VBScript,所以采用我们非常熟悉的 VB 语法。

② 公司特征: 由 Microsoft 公司推出。

③ 难易程度: 由于采用 Basic 语法,只要熟悉 Basic 语言,有点 HTML 基础,要学习掌握 ASP 是很简单的事情。

④ 代码隐藏性: 不需要编译,直接运行,所以代码可视。

⑤ 编写工具: 任何文本编辑器都可以进行编辑。

⑥ 封装性: 能很好地结合 MS 的 COM+ 技术,可以将比较复杂的事务处理工作封装在 COM+ 中,而且能非常好地进行调用。

⑦ 扩充性: 由于 MS 本身的 ActiveX 技术具有无限可扩充性,所以能很好地结合其他语言编写的组件。

⑧ 数据库: 采用 MS 的 ODBC 接口技术,所以,几乎所有的数据库都可以结合。

⑨ 平台性: 遗憾的是,对于我们目前流行的两种最主流的网络操作系统(UNIX/Linux 和 Windows),它却只能在 Windows 平台上受到很好的支持。

(2) 商业角度

① 平台成本: 可以选择 Windows Server+IIS+ASP+MS SQL Server 方案,因为这些都是 ASP“母”公司的产品,所以,从性能方面考虑,对于 ASP 肯定是最佳的方案,但這些产品中,几乎都要支付一定的费用,对于一个小型公司来说,建设一个 ASP 的网站是比较“贵”的。

② 人工成本：ASP 技术简单，ASP 人才几乎到处可见，所以，开发一个 ASP 网站、维护一个 ASP 网站，单从人工成本上来说，是不需要太多费用的。

2. JSP

(1) 技术角度

① 简介：JSP 全名 Java Server Pages，采用 Java 语法，Java 体系的任何东西都是需要 JDK 支持的，同样，JSP 也离不开 JDK。

② 公司特征：由 Sun 公司推出。

③ 难易程度：由于采用 Java 语法，而 Java 具有高度的面向对象和灵活性，所以，比 Basic 稍微要难点。

④ 代码隐藏性：要编译成 Servlet，在服务器端运行，所以代码不可视。

⑤ 编写工具：任何文本编辑器都可以进行编辑，但是，却需要 JDK 先编译好。

⑥ 封装性：能很好地结合 JavaBean 技术，可以将复杂的事务处理工作封装在 Bean 中，而 JSP 能非常好地进行调用。

⑦ 数据库：采用 Java 结合数据库技术，即 JDBC 技术，也是一个统一的数据库接口技术。

⑧ 平台性：对于目前流行的两种最主流的网络操作系统（UNIX/Linux 和 NT/Windows 2000），都能得到很好的支持。

(2) 商业角度

① 平台成本：有很多可以考虑的非常优秀的方案，但是，这些“非常优秀”的方案中，其中的价格也是非常昂贵的。当然，也可以采用一些不需要支付任何费用的方案，如 Linux+Tomcat+JSP+MySQL 等。

② 人工成本：从目前来看，由于 JSP 的诞生没有多长时间，而且，要真正将 Java 体系掌握清楚，并不是短时间就能做到的，所以，目前 JSP 人才相对来说，应该要比 ASP 和 PHP 人才少得多，从商业角度分析，成本也要贵不少。

3. PHP

(1) 技术角度

① 语法：PHP，采用类似 C 的语法。

② 难易程度：由于采用 C 语言语法，所以要学习掌握 PHP 也是比较简单的事情。

③ 代码隐藏性：不需要编译，直接运行，所以代码可视。但是，可以通过 Zend 的编译器将其代码加密处理，以隐藏源代码。

④ 编写工具：任何文本编辑器都可以进行编辑。

⑤ 封装性：能结合 MS 的 COM+ 技术，也能结合 JavaBean，将某些复杂的事务处理工作封装在 COM+ 和 JavaBean 中，但是，性能当然没有像 ASP 结合 COM+ 和 JSP 结合 JavaBean 那么“原版”，也就是速度和性能肯定没有后者好。

⑥ 公司特征：PHP 是开源的，所以，你可以得到它的源代码，并可以重新编译，甚至加入自己的特征。

⑦ 数据库：对于目前流行的数据库，几乎都有支持。但是，它却不是像 ASP、JSP 那样有个统一的接口，对于每种数据库几乎都有一个不同的接口。

⑧ 平台性：对于目前流行的两种最主流的网络操作系统（UNIX/Linux 和 Windows），它都可以得到很好的支持，而且根本不用修改任何代码。

（2）商业角度

① 平台成本：可以考虑 Linux+Apache+PHP+MySQL 方案，因为这个方案是 PHP 的最佳选择。而且令人兴奋的是，这些都是开放源代码的产品。也就是说，可以不向任何人支付任何费用就可以做到的。

② 人工成本：从目前来看，由于 PHP 目前也是非常流行的，所以，PHP 的人才也非常多。而且，开发速度也是非常快，所以从这个方面考虑，成本也是不高的。

4. 前景分析

对于比较大型的网站，比如对事务处理和负载均衡要求比较高的站点，采用 JSP 和 ASP 的比较多。从成本上考虑，比较经济的站点采用 PHP 应该是最好的选择。

由于三种语言各有自己的长处，所以，都有相当的支持者，在今后相当一段时间内，都不会被对方所淘汰。

1.3 Java、Servlet 和 JSP 的联系

本节将要简单介绍 Java、Servlet 和 JSP 的发展及联系。

1. Java

Java 是一种可以撰写跨平台应用软件的面向对象的程序设计语言，是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台（即 Java SE、Java EE、Java ME）的总称。Java 技术具有卓越的通用性、高效性、平台移植性和安全性，广泛应用于个人 PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网，同时拥有全球最大的开发者专业社群。在全球云计算和移动互联网的产业环境下，Java 更具备了显著优势和广阔前景。

与传统程序不同，Sun 公司在推出 Java 之际就将其作为一种开放的技术。全球数以万计的 Java 开发公司被要求所设计的 Java 软件必须相互兼容。“Java 语言靠群体的力量而非公司的力量”是 Sun 公司的口号之一，并获得了广大软件开发商的认同。这与微软公司所倡导的注重精英和封闭式的模式完全不同。Sun 公司对 Java 编程语言的解释是：Java 编程语言是个简单、面向对象、分布式、解释性、稳健、安全与系统无关、可移植、高性能、多线程和动态的语言。Java 平台是基于 Java 语言的平台，这样的平台目前非常流行。因此，微软公司推出了与之竞争的 .NET 平台以及模仿 Java 的 C# 语言。

2. Servlet

Servlet 是一种服务器端的 Java 应用程序，具有独立于平台和协议的特性，可以生成动态的 Web 页面。它担当客户请求（Web 浏览器或其他 HTTP 客户程序）与服务器响应（HTTP 服务器上的数据库或应用程序）的中间层。Servlet 是位于 Web 服务器内部的服务器端的 Java 应用程序，与传统的、从命令行启动的 Java 应用程序不同，Servlet 由 Web 服务器进行加载，该 Web 服务器必须包含支持 Servlet 的 Java 虚拟机。

服务器上需要一些程序,常常是根据用户输入访问数据库的程序。这些通常是使用公共网关接口(Common Gateway Interface,CGI)应用程序完成的。

在通信量大的服务器上,Java Servlet 的优点在于,它们的执行速度更快于 CGI 程序。各个用户请求被激活成单个程序中的一个线程,而不必创建单独的进程,这意味着服务器端处理请求的系统开销将明显降低。

Java Servlet 是 Java 语言的一部分,提供了用于服务器编程的 API。用 Java Servlet 编写的 Java 程序,称为一个 Servlet。Servlet 通过 HTML 与客户交互信息。Servlet 的最大缺点是,不能有效地管理页面的逻辑部分和页面的输出部分,导致 Servlet 代码非常混乱,用 Servlet 来管理网站变成一件很困难的事情。为了克服 Servlet 的缺点,SUN 公司用 Java Servlet 作为基础,推出了 Java Server Pages,即 JSP。

3. JSP

JSP 是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准。JSP 提供了 Servlet 的几乎所有好处,当一个客户请求一个 JSP 页面时,JSP 引擎根据 JSP 页面生成一个 Java 文件,即一个 Servlet。用 JSP 支持 JavaBean 这一特点,可以有效地管理页面的逻辑部分和页面的输出部分。另外,JSP 也可以和 Servlet 有效地结合,分离页面的逻辑部分和页面的输出部分。JSP 技术有点类似 ASP 技术,它是在传统的网页 HTML 文件(*.htm、*.html)中插入 Java 程序段(Scriptlet)和 JSP 标记(Tag),从而形成 JSP 文件(*.jsp)。用 JSP 开发的 Web 应用是跨平台的,既能在 Linux 下运行,也能在其他操作系统上运行。

JSP 技术具有如下特点。

- (1) 一次编写,到处运行。除了系统之外,代码不用做任何更改。
- (2) 系统的多平台支持。基本上可以在所有平台上的任意环境中开发,在任意环境中进行系统部署,在任意环境中扩展。相比 ASP/.NET 的局限性是显而易见的。
- (3) 强大的可伸缩性。从只有一个小的 Jar 文件就可以运行 Servlet/JSP,到由多台服务器进行集群和负载均衡,Java 显示了巨大的生命力。
- (4) 多样化和功能强大的开发工具支持。这一点与 ASP 很像,Java 已经有了许多非常优秀的开发工具,而且许多可以免费得到,并且其中许多已经可以顺利地运行于多种平台之下。
- (5) 支持服务器端组件。Web 应用需要强大的服务器端组件来支持,开发人员需要利用其他工具设计实现复杂功能的组件供 Web 页面调用,以增强系统性能。JSP 可以使用成熟的 JavaBean 组件来实现复杂的商务功能。

1.4 JSP 工作原理

JSP 工作原理如图 1.1 所示。当服务器上的一个 JSP 页面被第一次请求执行时,服务器上的 JSP 引擎首先将 JSP 页面文件转译成一个 Java 文件,再将这个 Java 文件编译生成字节码文件,然后通过执行字节码文件响应客户的请求,而当这个 JSP 页面再次被请求执行时,JSP 引擎将直接执行这个字节码文件来响应客户,这也是 JSP 比 ASP 速度

快的一个原因。而 JSP 页面的首次执行往往由服务器管理者来执行。这个字节码文件的主要工作如下。

(1) 把 JSP 页面中普通的 HTML 标记符号(页面的静态部分)交给客户的浏览器负责显示。

(2) 执行“<%”和“%>”之间的 Java 程序段(JSP 页面中的动态部分),并把执行结果交给客户的浏览器显示。

(3) 当多个客户请求同一个 JSP 页面时,JSP 引擎为每个客户启动一个线程而不是启动一个进程,这些线程由 JSP 引擎服务器来管理,与传统的 CGI 为每个客户启动一个进程相比较,效率要高得多。

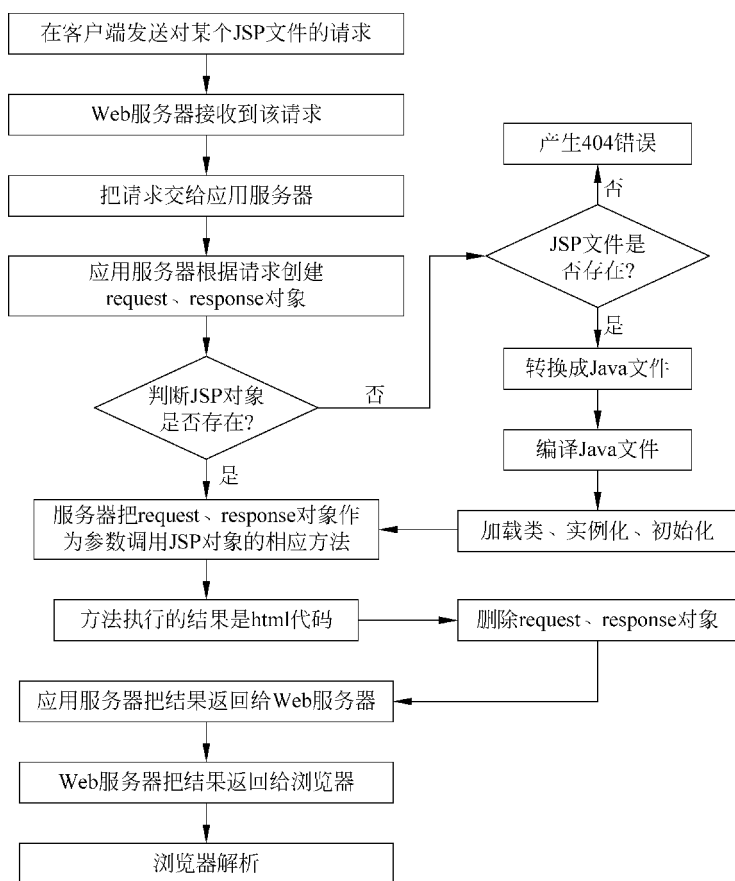


图 1.1 JSP 工作原理

当客户端向一个 JSP 页面发出请求时,Web Container(容器)将 JSP 转化成 Servlet 的源代码(只在第一次请求时),然后编译转化后的 Servlet 并加载到内存中执行,执行的结果 Response(响应)到客户端。JSP 只在第一次执行的时候会转化为 Servlet,以后每次执行 Web 容器都是直接执行编译后的 Servlet,所以 JSP 和 Servlet 只是在第一次执行的时候不一样,JSP 慢一点,以后的执行都是相同的。

下面是被请求的 JSP 文件(sum.jsp)及其在 JSP 引擎里生成的对应的 Java 文件,并

把 JSP 引擎交给客户端显示的内容做了“***”注释。

sum.jsp 源文件如下：

```
<%@ page contentType="text/html;charset=GB2312" %>
<HTML>
<BODY BGCOLOR=cyan>
<FONT Size=1>
<P>这是一个简单的 JSP 页面
    <% int i, sum=0;
        for(i=1;i<=100;i++)
        { sum=sum+i;
        }
    %>
<P> 1 到 100 的连续和是:
<BR>
    <%=sum %>
</FONT>
</BODY>
</HTML>
```

sum.jsp 在 JSP 引擎里生成的对应的 Java 文件如下：

```
package org.apache.jsp;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
import org.apache.jasper.runtime.*;
public class first1 $ jsp extends HttpJspBase {
    static {
    }
    public first1 $ jsp() {
    }
    private static boolean _jspx_inited = false;
    public final void _jspx_init() throws org.apache.jasper.runtime.JspException {
    }
    public void _jspService(HttpServletRequest request, HttpServletResponse response)
    throws java.io.IOException, ServletException {
        JspFactory _jspxFactory = null;
        PageContext pageContext = null;
        HttpSession session = null;
        ServletContext application = null;
        ServletConfig config = null;
        JspWriter out = null;
        Object page = this;
        String _value = null;
        try {
            if (_jspx_inited == false) {
                synchronized (this) {
                    if (_jspx_inited == false) {
                        _jspx_init();
                        _jspx_inited = true;
                    }
                }
            }
        }
```

```

    }
    _jspxFactory = JspFactory.getDefaultFactory();
    response.setContentType("text/html;charset=GB2312");
    pageContext = _jspxFactory.getPageContext(this, request, response, "", true, 8192, true);
    application = pageContext.getServletContext();
    config = pageContext.getServletConfig();
    session = pageContext.getSession();
    out = pageContext.getOut();
    ( ***) out.write("\r\n<HTML>\r\n<BODY>\r\n<P>这是一个简单的 JSP 页面\r\n");
    int i, sum=0;
        for(i=1;i<=100;i++)
        { sum=sum+i;
        }
    ( ***) out.write("\r\n<P> 1 到 100 的连续和是: \r\n<BR>\r\n    ");
    ( ***) out.print(sum );
    ( ***) out.write("\r\n</BODY>\r\n<HTML>\r\n");
} catch (Throwable t) {
    if (out != null && out.getBufferSize() != 0)
        out.clearBuffer();
    if (pageContext != null) pageContext.handlePageException(t);
} finally {
    if (_jspxFactory != null) _jspxFactory.releasePageContext(pageContext);
}
}
}

```

1.5 实践任务：用记事本开发第一个 JSP 应用

1. 任务说明

编写一个简单的 HTML 文件——count.html, 这个 HTML 中包含一个 Form 表单, 这个表单可以接收用户输入的数字, 然后把这个数字发送到 Web 服务器 Tomcat 端的 helloworld.jsp。helloworld.jsp 根据接收的参数决定显示内容重复的次数, 如图 1.2 所示。

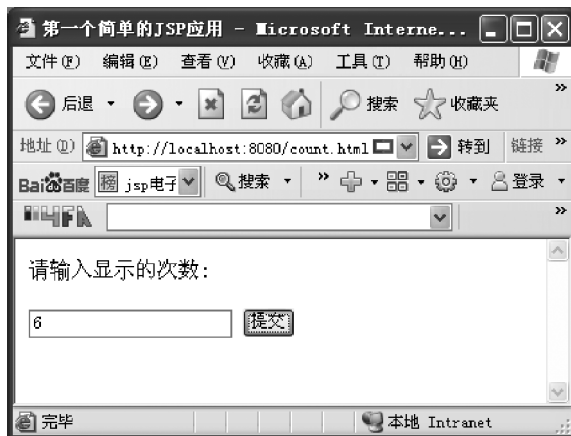


图 1.2 count.html 显示的结果