

高等院校信息技术规划教材

网络程序设计JSP

汪诚波 主 编
徐秀华 赵 怡 副主编
刘启玉 肖 辉 编 著

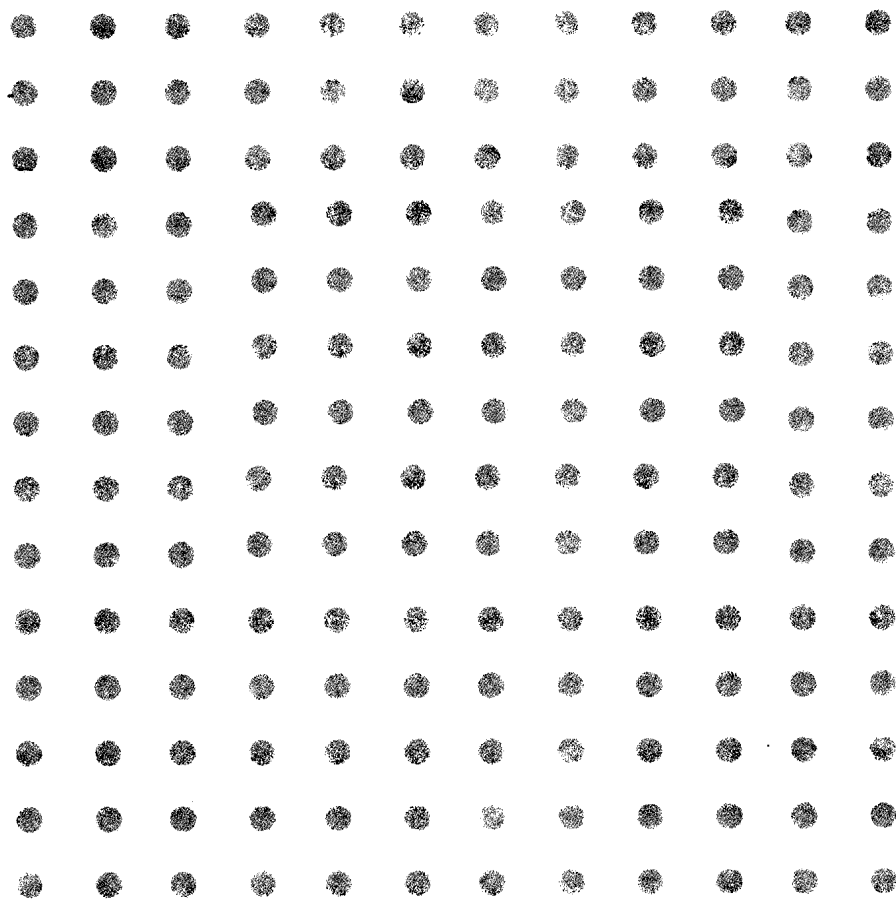


清华大学出版社

高等院校信息技术规划教材

网络程序设计JSP

汪诚波 主 编
徐秀华 赵 怡 副主编
刘启玉 肖 辉 编 著



清华大学出版社
北京

内 容 简 介

本书全面介绍网络程序设计 JSP 开发技术,内容包括 HTML 简介、JavaScript 简介、CSS 简介、JSP 基本语法、JSP 内置对象、JavaBean、JSP 文件操作、JSP 数据库开发、Servlet 技术等几方面的内容。

作为计算机应用技术类的教材,本书集成了程序设计技术、数据结构及算法、Java 程序设计、数据库技术、软件工程、软件体系结构等知识,重点是上述知识在 JSP 技术环境中的综合运用。

本书不仅适合作为高等学校应用型本科的教材,也适合自学者及工程技术人员参考使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

网络程序设计 JSP/汪诚波主编. —北京:清华大学出版社,2011.2

(高等院校信息技术规划教材)

ISBN 978-7-302-24569-8

I. ①网… II. ①汪… III. ①Java 语言—主页制作—程序设计—高等学校—教材

IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2011)第 009647 号

责任编辑:焦 虹

责任校对:梁 毅

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjjc@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 刷 者:三河市君旺印装厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:20.25

字 数:478 千字

版 次:2011 年 2 月第 1 版

印 次:2011 年 2 月第 1 次印刷

印 数:1~3000

定 价:30.00 元

产品编号:039172-01

前言

foreword

本书是作者在多年的教学与科研实践的基础上,按照普通高等本科院校“应用型人才”的培养目标和基本要求而编写的一本计算机应用技术类专业教材。本书在全面讲解网络程序设计的基本知识的同时,从工程实践出发,重点强调知识的实际运用能力。本教材摒弃了传统本科教材用抽象的表达式或者无实用价值的例子来解析软件工程理论的方法,也不采用高职教材中的最典型的“step”案例说明法,而是把软件工程理论、OOP 思想等偶于案例中,以更高的视野去审视、分析案例:一方面,通过对具有实用价值案例的学习,掌握基本概念、基本原理及技术规范,另一方面,通过对案例的分析,达到活用技术的目的。

技术是没有先进与落后的,只有合适与不合适之分。一个工程项目采用何种解决方案,是没有标准答案的。在网络程序设计(JSP)中,本身的技术规范及原理并不很难,但要掌握及灵活运用技术,且是一件不容易的事。问题的解决本身存在着一般规律和原则,如何用软件开发的一般规律和原则去分析案例,去理解案例中所运用的各项技术及规范,以达到掌握技术的目的,是本教材向读者要传递的基本思想。本教材采用的案例来源于已开发并应用的实际项目,对案例的分析很多直接来源于工程技术人员的现场讨论。本教材尽可能把各种解决方案及优缺点呈现在作者面前,务求读者从更高层次上来理解各项技术。

本教材全面介绍了 JSP 技术,重点讲解了三个内容:JavaBean 技术、Servlet 技术和 JDBC 数据库应用技术。对于较难理解的问题,本教材采用“现场→抽象→技术方案→特点分析→选择方案”的组织方式。本书是计算机应用技术类专业教材,需具有一定的计算机专业基础知识。

由于作者水平有限,加之时间紧张,难免存在缺点和错误,恳请广大读者给予批评指正。

作者

2010 年 11 月

目录

Contents

第 1 章 JSP 网络编程技术概述	1
1.1 Web 应用开发	1
1.1.1 B/S 结构	1
1.1.2 B/S 结构的特点	2
1.1.3 B/S 结构的优缺点	2
1.2 动态网页技术	3
1.2.1 CGI 技术	3
1.2.2 ASP 技术	4
1.2.3 ASP.NET 技术	5
1.2.4 PHP 技术	7
1.2.5 Servlet/JSP 技术	8
1.2.6 Web 开发技术比较	11
1.3 JSP 技术原理	12
1.3.1 JSP 工作原理	12
1.3.2 JSP 工作环境	13
1.3.3 第一个 JSP 页面	14
1.4 本章小结	16
第 2 章 JSP 开发环境搭建	17
2.1 安装 JDK	17
2.2 安装和配置 Tomcat	18
2.2.1 安装 Tomcat	18
2.2.2 配置 Tomcat	19
2.3 安装和配置开发环境: Eclipse	22
2.3.1 安装 Eclipse	22
2.3.2 配置 Eclipse	24
2.3.3 在 Eclipse 环境下建立 Web 项目	26



2.4	安装数据库: SQL Server 2005	28
2.4.1	安装 Internet 信息服务(IIS)	29
2.4.2	安装 SQL Server 2005	29
2.5	本章小结	37
第 3 章	基本语法	38
3.1	JSP 页面结构	38
3.2	注释	39
3.3	脚本元素	40
3.3.1	声明	40
3.3.2	表达式	42
3.3.3	脚本小程序	43
3.3.4	关于 JSP 中 Java 代码的总结	45
3.4	指令标签	50
3.4.1	page 指令标签	50
3.4.2	include 指令标签	53
3.4.3	taglib 指令标签	54
3.5	动作元素	54
3.5.1	<jsp:include>	55
3.5.2	<jsp:forward>	56
3.5.3	<jsp:param>	58
3.5.4	<jsp:plugin>、<jsp:params>和<jsp:fallback>	59
3.6	本章小结	60
第 4 章	JSP 内置对象	61
4.1	内置对象概述	61
4.2	request 对象	61
4.2.1	request 对象的主要方法	62
4.2.2	request 对象的常用技术	64
4.3	response 对象	68
4.3.1	response 对象的主要方法	68
4.3.2	response 对象的常用技术	69
4.4	session 对象	72
4.4.1	session 对象的主要方法	72
4.4.2	session 对象的常用技术	73
4.5	out 对象	78
4.5.1	out 对象的主要方法	78

4.5.2	out 对象的常用技术	79
4.6	page 对象	80
4.7	application 对象	80
4.7.1	application 对象的主要方法	80
4.7.2	application 对象的常用技术	80
4.8	exception 对象	82
4.8.1	exception 对象简介	82
4.8.2	exception 对象的主要方法	82
4.8.3	exception 对象示例	82
4.9	pageContext 对象	84
4.9.1	pageContext 对象简介	84
4.9.2	pageContext 对象的主要方法	84
4.10	config 对象	85
4.10.1	config 对象简介	85
4.10.2	config 对象的主要方法	85
4.11	本章小结	85
第 5 章	JavaBean 组件	86
5.1	JavaBean 的基本概念	86
5.2	JavaBean 的设计与开发	87
5.2.1	JavaBean 的规则	87
5.2.2	设计一个简单的 JavaBean	87
5.2.3	在 JSP 中使用 JavaBean	89
5.3	进一步探讨 JavaBean 在 JSP 中的使用	92
5.3.1	利用 JSP 标签设置或获取 JavaBean 属性	92
5.3.2	JavaBean 的作用域范围	94
5.4	综合案例购物车设计	96
5.4.1	购物车的业务需求分析	96
5.4.2	各种技术方案分析	97
5.4.3	具体实现	98
5.5	本章小结	105
第 6 章	Servlet 基础	106
6.1	Servlet 技术简介	106
6.1.1	Servlet 的历史及技术特点	106
6.1.2	Servlet 的主要功能、运行过程及生命周期	107
6.1.3	开发部署一个简单的 servlet	109



6.2	Servlet 体系的常用类和接口	111
6.2.1	常用类和接口	111
6.2.2	Servlet 的配置	114
6.3	Servlet 技术的应用	115
6.3.1	Servlet 过滤器技术	116
6.3.2	Servlet 监听器技术	118
6.4	本章小结	122
第 7 章	Servlet 与 MVC 开发模式	123
7.1	MVC 开发模式	123
7.1.1	MVC 简介	123
7.1.2	Web 项目中的 MVC	124
7.1.3	关于 MVC 的总结	125
7.2	应用案例	126
7.2.1	一个简单的例子	126
7.2.2	购物车 MVC 代码重构	128
7.3	文章发布系统	133
7.3.1	需求分析	133
7.3.2	代码实现	134
7.4	本章小结	138
第 8 章	JDBC 技术	139
8.1	JDBC 原理概述	139
8.1.1	JDBC 基本概念	139
8.1.2	JDBC 驱动程序及安装	140
8.1.3	一个简单的 JDBC 应用实例	141
8.2	JDBC 常用接口、类介绍	142
8.2.1	Driver 接口	142
8.2.2	DriverManager 类	143
8.2.3	Connection 接口	144
8.2.4	Statement	146
8.2.5	ResultSet 结果集	152
8.3	使用 JDBC 元数据	157
8.3.1	DatabaseMetaData 的使用	157
8.3.2	ResultSetMetaData 的使用	158
8.3.3	ParameterMetaData 的使用	159
8.4	使用 JDBC 连接不同数据库	160

8.5 本章小结	162
第9章 JDBC 在 JSP 中的应用	163
9.1 数据库访问层的设计与实现	163
9.1.1 连接管理器(DBConnection)的设计与实现	163
9.1.2 数据库服务类 DBOperation 的设计与实现	164
9.2 在 MVC 模式中应用 JDBC	168
9.2.1 登录过程的设计与实现	168
9.2.2 书籍展示的设计与实现	170
9.2.3 注册过程的设计与实现	174
9.2.4 成批数据的录入与修改	177
9.3 分页处理的设计与实现	181
9.3.1 各种实现方案比较	181
9.3.2 M 层设计	182
9.3.3 C 层设计	183
9.3.4 V 层设计	186
9.3.5 代码结构及运行结果	187
9.4 本章小结	187
第10章 数据库高级应用	188
10.1 JNDI 数据源与连接池技术	188
10.1.1 连接池技术简介	188
10.1.2 数据源与 JNDI 技术	189
10.1.3 配置数据源与连接池	191
10.1.4 在 JSP 中应用数据源技术	192
10.2 使用存储过程	194
10.2.1 存储过程的基本概念	194
10.2.2 利用存储过程解决分页问题	194
10.3 本章小结	196
第11章 JSP 开发中的常用技术	197
11.1 中文处理技术	197
11.1.1 中文乱码的由来	197
11.1.2 解决方案	198
11.2 文件处理技术	199
11.2.1 文件读写的基本概念	200
11.2.2 文件上传、下载的基本概念	206



11.2.3 Smartupload 组件介绍	211
11.3 本章小结	214
第 12 章 Web 页面设计技术简介	215
12.1 HTML 简介	215
12.1.1 HTML 的基本结构	216
12.1.2 HTML 文件头	216
12.1.3 HTML 文件主体	218
12.1.4 文字与段落	218
12.1.5 列表	222
12.1.6 超链接	223
12.1.7 表格	224
12.1.8 表单	227
12.2 CSS 简介	232
12.2.1 CSS 基本语法	232
12.2.2 CSS 分类	233
12.2.3 CSS 的冲突	235
12.2.4 CSS 的属性单位	235
12.2.5 CSS 属性	237
12.3 JavaScript 简介	241
12.3.1 JavaScript 基本语法	242
12.3.2 JavaScript 的内置对象	249
12.3.3 浏览器对象	252
12.4 本章小结	256
第 13 章 综合应用案例——网上书店	257
13.1 需求分析	257
13.2 数据库表设计	257
13.3 模型层设计	259
13.3.1 DAO 层设计	259
13.3.2 业务层设计	265
13.3.3 通用组件的设计	270
13.4 前台主页面设计	272
13.4.1 主页面的各部分布局	273
13.4.2 主页面的具体设计	273
13.5 网上书店的具体实现	282
13.5.1 主要业务流程分析	282

13.5.2 主要的页面及 Servlet 设计	282
13.5.3 用户管理模块的具体实现	283
13.5.4 图书搜索及展示的具体实现	291
13.5.5 购物过程的具体实现	299
13.6 本章小结	306
参考文献	307

第 1 章

chapter 1

JSP 网络编程技术概述

随着 Web 时代的到来,诞生了许多 Web 上的新兴应用及其相关软件,而许多传统的应用软件也往往需要改造成为 Web 上的应用软件。本章主要介绍 Web 上的应用软件开发的原理和相关概念,然后对几种主要的 Web 编程技术进行逐一介绍并加以对比,最后重点介绍 JSP 网络编程技术,并给出一个简单的实例。

1.1 Web 应用开发

1.1.1 B/S 结构

Web 上的应用软件主要基于 B/S 结构(Browser/Server 结构),即浏览器和服务器结构。它是随着 Internet 技术的兴起,对 C/S 结构(Client/Server 结构)的一种变化或者改进的结构。在这种结构下,用户工作界面通过 WWW 浏览器来实现,极少部分事务逻辑在前端(浏览器,Browser)实现,主要事务逻辑在服务器(Server)端实现,并根据需要,将服务器端分解为数据访问层和业务逻辑层,形成所谓的三层(3-tier)架构。如图 1-1 所示。

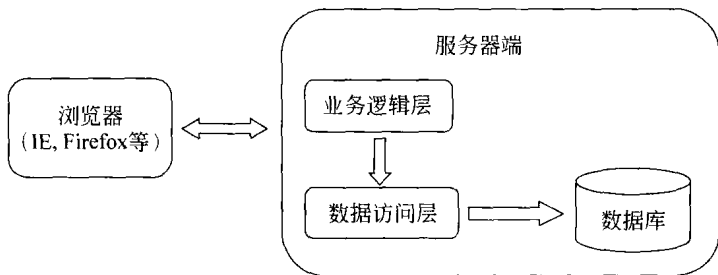


图 1-1 B/S 系统的三层结构

其中浏览器作为用户的工作界面,主要用于显示数据和接收用户的输入,为用户提供交互式的操作界面,可以实现简单的校验逻辑。业务逻辑层是整个系统中最核心的部分,主要集中在业务规则的制定、业务流程的实现等与业务需求有关的系统设计。即它与系统所应对的领域(Domain)逻辑有关,因此也可将业务逻辑层称为领域层。数据访问

层有时也称为持久层,其功能主要是负责数据库的访问,可以访问数据库系统、二进制文件、文本文档或是 XML 文档。

1.1.2 B/S 结构的特点

与 C/S 结构相比,B/S 结构具有明显不同的特点。

1. 硬件环境不同

C/S 一般建立在专用的局部网络上,局域网之间通过专门服务器提供连接和数据交换服务,客户机相对固定不变,且具有相同的操作系统;B/S 建立在广域网之上的,不必是专门的网络硬件环境,客户机只要有操作系统和浏览器就行,硬件适应性要求较高。

2. 处理模式不同

C/S 每台客户机安装相同的客户端软件,通过网络协议连接到服务器端软件。客户端软件不但能够完成数据显示和界面控制,也能完成一些较为专业的、复杂的应用业务逻辑。

B/S 大大简化了客户端,只要装上操作系统、网络协议软件以及浏览器即可。这时客户机成为瘦客户机,而服务器则集中了所有的应用逻辑。

3. 构件重用不同

C/S 软件需要考虑系统的整体性,构件的重用性较低;B/S 一般采用多重结构,要求构件相对独立的功能,具有较好的重用性。

4. 交互性不同

交互性强是 C/S 固有的一个优点。在 C/S 中,客户端有一套完整的应用程序,在出错提示、在线帮助等方面都有强大的功能,客户端能够保持数据的所有状态。B/S 结构中,虽然浏览器由 JavaScript 等脚本语言提供了一定的交互能力,但是用户感觉远不如 C/S 结构,更重要的是浏览器不能保持数据状态,需要手工来维持。

1.1.3 B/S 结构的优缺点

B/S 结构的软件具有如下优点:

(1) 分布性特点: B/S 结构的客户端是普通的浏览器,所以只要用户计算机安装了浏览器,就可以适用 B/S 结构的软件。因此,B/S 结构的软件非常适用于客户端分散在各地的企业级应用系统。

(2) 可维护性好: 目前,软件系统的改进和升级越来越频繁,在 C/S 结构中,客户端软件一旦改动,则需要对所有用户的客户端软件重新发布,因此软件维护和升级十分不便。与 C/S 处理模式相比,B/S 的客户端是普通的浏览器,开发、维护等几乎所有工作都

集中在服务器端。当企业对网络应用进行升级时,只需更新服务器端的软件就可以了,这就减轻了异地用户系统维护与升级的成本。

(3) 系统性能稳定: B/S 结构的软件,其客户端不需要维护,而服务器端通常由高性能的服务器系统承担,性能稳定。当客户数增加时,只需在服务器端增加硬件设备,利用负载均衡和网络优化技术就能够很好地应付。

与 C/S 结构相比,B/S 结构软件的缺点也很明显。这主要是客户端的浏览器软件造成的,可以归结为以下两点:

(1) 浏览器功能简单: 浏览器作为 B/S 结构软件的客户端,主要用于简单的数据验证和数据显示功能,复杂数据处理一般都交给服务器完成,客户端和服务器的交互过程比较长,容易引起系统性能的下降。

(2) 浏览器兼容性问题: 目前浏览器的种类较多,不同浏览器存在兼容性问题,因此 B/S 结构软件在设计时需要处理浏览器的兼容性问题,否则容易使显示和操作异常。

1.2 动态网页技术

在 Web 网络环境下运行应用软件,需要解决用户和服务端的动态交互问题。HTML 技术是目前网络上应用最为广泛的语言,也是构成网页文档的主要语言。但是 HTML 页面不具备交互能力,其页面是静态的,包含的数据是固定的,无法对用户的不同请求给出正确的反应。为了解决用户和服务端端的动态交互问题,业界提出了动态网页技术,它具有以下几个特点。

(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 发展的大趋势。

(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。

除了早期的 CGI 外,目前主流的动态网页技术有 ASP、PHP、JSP 等,下面分别介绍这些动态网页技术。

1.2.1 CGI 技术

在早期,尽管 WWW 服务器不具有动态交互能力,但是它允许用户编写扩展应用程序来扩展服务器的功能,CGI(Common Gateway Interface,公用网关接口)就是这样一种标准扩展技术,它是外部扩展应用程序与 WWW 服务器交互的一个标准接口。按照 CGI 标准编写的外部扩展应用程序实际是运行在 WWW 服务器上的一个组件,可以处理用户通过浏览器输入的工作数据,完成浏览器与服务器的交互操作。其工作过程可以描述为: 首先用户可以在客户端指定请求的 CGI 应用程序的 URL,服务器收到请求后,会启动一个新的进程执行 CGI 应用程序,CGI 应用程序处理用户的输入数据后将结果格式转化为 HTML 网页形式返回到服务器,再由服务器返回到客户端。

CGI 应用程序可以由大多数编程语言编写,如 Perl、C\C++、Delphi、Visual Basic 和 Java 等。其中 Perl 语言的前身是属于 UNIX 的高级语言,因具有强大的字符串处理能力而成为编写 CGI 程序的首选。以下是 perl 语言编写的 CGI 例子。

例 1-1 CGI 程序示例。

```
#!/usr/bin/perl-w
use CGI
{
    my $q=new CGI
    print $q->header()
    $q->start_html("Hello perl world!")
    $q->h1('hello perl world')
    $q->end_html()
}
```

输出结果为:

Hello Perl world

由于每次用户提出动态内容请求时,服务器都需要启动一个新的进程为 CGI 服务,因此当用户数增加时,需要创建许多进程,消耗大量的系统资源,这样很容易造成系统整体性能的下降。同时由于 CGI 程序和网页 HTML 文档是完全分开编写和运行的,因此 CGI 程序的维护和编写都比较困难。

1.2.2 ASP 技术

ASP 是 Active Server Page 的缩写,意为“动态服务器页面”,是一个基于 Web 服务器端的开发技术,利用它可以产生和执行动态的、互动的、高性能的 Web 应用程序。ASP 是微软公司开发的代替 CGI 脚本程序的一种应用,采用脚本语言 VBScript 作为开发语言,借助于 COM+ 技术,几乎可以实现所有 C/S 应用程序的功能。另外,ASP 可通过

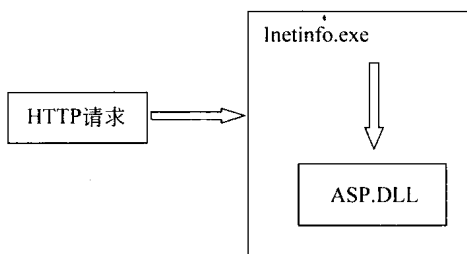


图 1-2 ASP 工作原理

ADO(ActiveX Data Object, 微软公司提出的一项高效访问数据库的技术)实现对各类数据库的访问。ASP 技术由于其语法简单、功能实用,再加上微软公司的大力整合和支持,在 20 世纪 90 年代成为了 Web 应用开发的主流技术之一。

ASP 技术的运行机制如图 1-2 所示。

当用户请求访问一个以 ASP 为扩展名的页面时,这个 HTTP 请求首先被 Inetinfo.exe 进程(它实际上就是 WWW 服务进程)所截获,而后它会将这个请求转交给 ASP.DLL 进程,ASP.DLL 进程负责解释执行这个 ASP 页面,然后将解释后的结果返回给客户端浏览器。

1. ASP 的优点

- 简单易学: Visual Basic 语言简单,且拥有非常广泛的开发群体,在微软公司非常完善的联机文档的帮助下,很容易掌握 ASP 技术。
- 运行简单: ASP 的运行环境非常简单,只需在 Windows 平台上安装好 IIS 即可,特别在一些 Windows Server 的操作系统下,本身就已经集成了 IIS 环境。
- 功能强大: 由于可以和 HTML 页面结合在一起,因此编写代码较为方便;而 COM+ 和 ADO 技术的支持,也使得 ASP 技术能开发出功能强大的 Web 应用程序。

2. ASP 的缺点

- 弱类型语言: ASP 采用脚本语言 Visual Basic 作为开发语言,是一种弱类型语言,需要解释运行,速度较慢。
- 安全性不高: ASP 需要 Windows 和 IIS 作为运行环境,安全性不如 Linux 和 UNIX 平台,容易受到攻击。
- 维护困难: ASP 是脚本语言和 HTML 语言混合编程,容易产生代码混乱,逻辑不清晰的问题,随着程序复杂性的增加,代码维护也比较困难。

3. ASP 示例

通常情况下,ASP 文件包含 HTML 标签,类似 HTML 文件。不过,ASP 文件也可包含服务器端脚本,这些脚本被`<%`和`%>`包围起来。服务器脚本在服务器端执行,可包含合法的表达式、语句或者运算符。下面给出一个简单的 ASP 示例。

例 1-2 ASP 示例。

```
<html>
  <body>
    <%
      response.write("Hello ASP World!")
    %>
  </body>
</html>
```

输出结果为:

Hello ASP world!

1.2.3 ASP.NET 技术

现在 ASP 技术已经被微软公司抛弃。2002 年以后,微软公司提出了全新的 ASP.NET,虽然名字也包含“ASP”,但是两者的编程模式完全不同。ASP.NET 是 Microsoft.net 的一部分,作为战略产品,它不仅是 ASP 的下一个版本,还提供了一个统



一的 Web 开发模型,其中包括开发人员生成企业级 Web 应用程序所需的各种服务。ASP.NET 的语法在很大程度上与 ASP 兼容,同时它还提供一种新的编程模型和结构,可生成伸缩性和稳定性更好的应用程序,并提供更好的安全保护。用户可以通过在现有 ASP 应用程序中逐渐添加 ASP.NET 功能,随时增强 ASP 应用程序的功能。

ASP.NET 是一个已编译的基于 .NET 的技术环境,可以用任何与 .NET 兼容的语言(包括 Visual Basic、.NET 和 C#)协同开发应用程序。另外,任何 ASP.NET 应用程序都可以使用整个 .NET Framework。开发人员可以方便地获得这些技术的支持,其中包括托管的公共语言运行库环境、类型安全、继承等。

微软公司为 ASP.NET 设计了这样一些策略:易于写出结构清晰的代码,代码易于重用和共享,可用编译类语言编写等,其目的是让程序员更容易开发出 Web 应用,满足向 Web 转移的战略需要。

与 ASP 相比,ASP.NET 具有明显的优势。

(1) 程序代码和网页内容分离,使得开发和维护简单方便。Code-Behind 技术将程序代码和 HTML 标记分离在不同的文件中。通过引入服务器端空间,并加入事件的概念,改变了脚本语言编写模式。

(2) 语言支持能力大大提高。ASP.NET 支持完整的 Visual Basic,而不是 VBScript 脚本语言,此外还支持面向对象的 C# 和 C++ 语言。

(3) 执行效率大幅提高。ASP.NET 是编译执行的,比起 ASP 的解释执行速度快了很多,并且提供了快速存取(Caching)的能力。

(4) 易于配置。通过纯文本文件就可完成对 ASP.NET 的配置,配置文件可在应用程序运行时,进行上传和修改,无须重启服务器。也没有 metabase 和注册方面的难题。

(5) 更高的安全性。改变了 ASP 单一的基于 Windows 的身份认证方式,增加了 Forms 和 Passport 两种身份认证方式。

ASP.NET 不完全兼容早期的 ASP 版本,所有大部分旧的 ASP 代码需要进行修改才能在 ASP.NET 技术环境下运行。为了解决这个问题,ASP.NET 使用了一个新的文件后缀".aspx"。这样,就使 ASP.NET 应用程序与 ASP 应用程序能够一起运行在同一个服务器上。

例 1-3 ASP.NET 示例。

```
<script runat="server">
Sub Page_Load
    response.write("Hello ASP.NET World!")
End Sub
</script>
<html>
    <body>
        </body>
</html>
```