

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

Web应用程序开发教程—— ASP.NET + SQL Server

Web Application Development——
ASP.NET+SQL Server

岳学军 李晓黎 主编

- 体现作者多年的Web应用程序开发经验
- 结合大量实用技巧，重点突出，便于灵活掌握
- 提供典型应用实例与上机实验，分析详细，实用性强



精品系列

 人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

Web应用程序开发教程—— ASP.NET + SQL Server

Web Application Development——
ASP.NET+SQL Server

岳学军 李晓黎 主编



精品系列

人民邮电出版社

北京

图书在版编目 (CIP) 数据

Web应用程序开发教程 : ASP.NET+SQL Server / 岳学军, 李晓黎主编. — 北京 : 人民邮电出版社, 2009. 11

21世纪高等学校计算机规划教材
ISBN 978-7-115-21506-2

I. ①W… II. ①岳… ②李… III. ①主页制作—程序设计—高等学校—教材②关系数据库—数据库管理系统, SQL Server—高等学校—教材 IV. ①TP393.092
②TP311.138

中国版本图书馆CIP数据核字 (2009) 第187554号

内 容 提 要

ASP.NET+SQL Server 是开发 Web 应用程序的经典组合, 被国内外众多网站广泛采用, 具有很强的实用性。本书首先系统介绍了 ASP.NET 程序设计和 SQL Server 2000 数据库管理的基础知识, 然后结合几个使用 ASP.NET+SQL Server 2000 开发 Web 应用程序的实例 (包括用户登录和身份验证、文件上传、发送电子邮件、图文验证、在线教育系统、网上书城系统等), 全面介绍了用 ASP.NET 和 SQL Server 2000 开发 Web 应用程序的方法和技巧。

本书既可以作为高等院校“Web 应用程序设计”课程的教材, 也可作为 Web 应用程序开发人员的参考用书。

21 世纪高等学校计算机规划教材

Web 应用程序开发教程——ASP.NET+SQL Server

◆ 主 编 岳学军 李晓黎

责任编辑 邹文波

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京铭成印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 27.75

字数: 731 千字

印数: 1—3 000 册

2009 年 11 月第 1 版

2009 年 11 月北京第 1 次印刷

ISBN 978-7-115-21506-2

定价: 42.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154



前言

互联网技术的不断发展和普及已经改变了人们的工作和生活习惯,很多人希望能够通过互联网足不出户地满足自己的需求,电子商务已经成为许多企事业单位的业务发展方向。因此,如何构建互联网站、开发 Web 应用程序已经成为当前的热门技术之一。高校的许多专业都开设了相关的课程。

开发 Web 应用程序必须了解两部分内容,即前台的开发工具和后台的数据库,本书选择了这一领域中的经典组合 ASP.NET + SQL Server,使读者能够掌握最实用的开发技术。

ASP.NET 是基于 .NET Framework 技术的新一代 Web 应用程序开发系统,它支持 Visual Basic.NET 和 Visual C# 等高级语言,因此它的功能比仅支持 VBScript 和 JavaScript 等脚本语言的 ASP 要强大得多。本书采用非常流行的 Visual C# 作为 ASP.NET 的开发语言,并在书中详细介绍了 Visual C# 语言的基本使用方法。

编者在多年开发 Web 应用程序和研究相关课程教学的基础上编写了本书。全书内容分为 4 个部分。第 1 部分介绍 ASP.NET 程序设计基础,由第 1~7 章组成,全面讲解了开发 Web 应用程序的基本流程、配置 ASP.NET 应用环境、Visual C# 语言、ASP.NET 服务器端控件、ASP.NET 内置对象和 ASP.NET AJAX 等;第 2 部分介绍 SQL Server 2000 数据库的管理和开发接口,由第 8~10 章组成,比较详尽地讲解了 Web 应用程序所必备的后台数据库管理及开发技术,读者无需再查阅其他数据库管理的参考资料;第 3 部分介绍非常实用的案例,由第 11~13 章组成,案例包括用户登录和身份验证、文件上传、发送电子邮件、图文验证、在线教育系统、网上书城系统等,这些案例具有很强的实用性,读者可以通过这些系统学习开发 Web 应用程序的过程和技术,也可以在实例的基础上稍加修改,独立使用;第 4 部分是附录,包括为了便于老师教学和学生实践设计的 8 个实验和 1 个综合性的大作业等。另外,本书前 10 章介绍的是基础知识,都配有相应的习题,以帮助读者理解所学习的内容,使读者加深印象、学以致用。

本书提供教学 PPT 课件、源程序文件、数据库脚本等,需要者可以登录人民邮电出版社教学服务与资源网(<http://www.ptpedu.com.cn>)免费下载。

本书在内容的选择、深度的把握上充分考虑初学者的特点,内容安排上力求做到循序渐进,不仅适合教学,也适合开发 Web 应用程序的各类人员自学使用。

由于编者水平有限,书中难免存在不足之处,敬请广大读者批评指正。

编 者

2009 年 8 月

目 录

第 1 部分 ASP.NET 程序设计基础

第 1 章 Web 应用程序设计与

开发概述.....2

1.1 Web 应用程序设计语言的产生与 发展.....2

1.2 Web 应用程序的基本开发流程和 工作原理.....3

1.3 当前比较流行的 Web 应用程序设计语言.....5

1.3.1 HTML 简介.....5

1.3.2 ASP 简介.....6

1.3.3 JSP 简介.....6

1.3.4 PHP 简介.....7

1.3.5 ASP.NET 简介.....7

习题.....7

第 2 章 ASP.NET 开发环境配置.....9

2.1 Web 服务器环境配置.....9

2.1.1 安装 IIS.....9

2.1.2 配置和管理 IIS.....10

2.1.3 申请和配置互联网站.....12

2.2 常用网页编辑工具——Dreamweaver.....13

2.2.1 Dreamweaver 主界面.....14

2.2.2 设置网页背景和颜色.....15

2.2.3 设置字体属性.....15

2.2.4 超链接.....16

2.2.5 图像和动画.....17

2.2.6 表格.....18

2.3 ASP.NET 项目开发基础.....19

2.3.1 创建 ASP.NET 项目.....19

2.3.2 ASP.NET 应用程序的目录和结构.....21

习题.....22

第 3 章 C#程序设计基础.....23

3.1 C#语言基础.....23

3.1.1 C#语言的基本特点.....23

3.1.2 .NET Framework 和 C#.....24

3.1.3 使用 Visual Studio 2005.....25

3.1.4 编写一个简单的 C#例子.....26

3.2 数据类型.....29

3.2.1 值类型.....30

3.2.2 引用类型.....33

3.2.3 类型转换.....36

3.3 常量和变量.....36

3.3.1 常量.....37

3.3.2 变量.....37

3.4 运算符和表达式.....38

3.4.1 算术运算符和算术表达式.....38

3.4.2 逻辑运算符和逻辑表达式.....39

3.4.3 递增递减运算符和递增递减表 达式.....39

3.4.4 关系运算符和关系表达式.....40

3.4.5 赋值运算符和赋值表达式.....40

3.5 流程控制语句.....41

3.5.1 选择控制语句.....42

3.5.2 循环控制语句.....45

3.5.3 跳转控制语句.....47

3.5.4 异常处理语句.....48

3.6 类和对象.....50

3.6.1 面向对象程序设计思想.....50

3.6.2 创建类.....51

3.6.3 构造函数和析构函数.....55

3.6.4 方法.....56

习题.....57

第 4 章 Web 窗体和常用服务器

控件.....59

4.1 Web 窗体.....59

4.1.1 Web 窗体中的 HTML 代码.....59

4.1.2 向 Web 窗体中添加控件.....60

4.1.3 Web 窗体中的隐藏文件.....	62	5.3 Context 对象.....	106
4.1.4 Web 窗体的事件模型.....	62	5.3.1 HttpContext 类的主要属性.....	106
4.1.5 Page 类.....	63	5.3.2 获取服务器上的时间.....	107
4.2 常用服务器端控件.....	63	5.3.3 获取服务器的名称.....	107
4.2.1 Label 控件.....	64	5.3.4 实现网页转向功能.....	108
4.2.2 TextBox 控件.....	65	5.4 Server 对象.....	109
4.2.3 Image 控件.....	67	5.4.1 HttpServerUtility 类的属性和方法.....	109
4.2.4 按钮相关控件.....	68	5.4.2 取得服务器的物理路径.....	109
4.2.5 HyperLink 控件.....	70	5.5 Trace 对象.....	110
4.2.6 CheckBox 控件和 CheckBoxList 控件.....	71	5.5.1 TraceContent 类的属性和方法.....	110
4.2.7 RadioButton 控件和 RadioButtonList 控件.....	74	5.5.2 使用 Trace 对象实现跟踪功能.....	110
4.2.8 AdRotator 控件.....	77	5.6 HttpRuntime 对象及其应用.....	112
4.2.9 Calendar 控件.....	79	5.6.1 HttpRuntime 对象的常用属性.....	113
4.2.10 Placeholder 控件.....	81	5.6.2 HttpRuntime 对象的应用实例.....	113
4.2.11 Panel 控件.....	82	习题.....	114
4.3 服务器验证控件.....	83	第 6 章 ASP.NET 应用程序配置和状态管理	115
4.3.1 RequireFieldValidator 控件.....	83	6.1 配置 ASP.NET 应用程序的工作环境.....	115
4.3.2 RangeValidator 控件.....	84	6.1.1 ASP.NET 配置系统概述.....	115
4.3.3 RegularExpressionValidator 控件.....	86	6.1.2 创建应用程序虚拟目录.....	116
4.3.4 CompareValidator 控件.....	89	6.1.3 ASP.NET 网站管理工具.....	117
4.3.5 ValidationSummary 控件.....	90	6.1.4 Web.config 文件的根元素.....	120
4.3.6 CustomValidator 控件.....	93	6.1.5 配置节处理程序声明区域.....	120
4.4 用户控件.....	95	6.1.6 配置节设置区域.....	121
4.4.1 添加用户控件.....	96	6.1.7 appSettings 元素.....	123
4.4.2 在 Web 窗体中使用用户控件.....	97	6.1.8 ConnectionStrings 元素.....	124
习题.....	97	6.1.9 ASP.NET 配置文件的继承关系.....	125
第 5 章 ASP.NET 对象	99	6.2 ASP.NET 状态管理.....	126
5.1 Response 对象.....	99	6.2.1 ASP.NET 的生命周期.....	126
5.1.1 输出到网页.....	99	6.2.2 Application 状态及其应用.....	128
5.1.2 网页转向.....	100	6.2.3 Session 状态及其应用.....	134
5.1.3 写入 Cookie 数据.....	100		
5.2 Request 对象.....	101		
5.2.1 读取网址的参数信息.....	101		
5.2.2 读取表单传递的数据.....	102		
5.2.3 读取和写入 Cookie 数据.....	103		
5.2.4 读取服务器端的环境变量.....	104		

6.2.4 ViewState 状态及其应用	138
6.2.5 Session、Application 和 ViewState 的比较	141
习题	142

第 7 章 ASP.NET AJAX 服务器

端编程	144
7.1 ASP.NET AJAX 基础	144
7.1.1 Ajax 的基本概念和特点	144
7.1.2 下载和安装 ASP.NET AJAX	146
7.2 使用 ScriptManager 控件	149
7.3 使用 UpdatePanel 控件实现网页局部 刷新	150

第 2 部分 SQL Server 2000 数据库管理与开发接口

第 8 章 SQL Server 2000 数据库 管理

8.1 常用数据库管理工具	162
8.1.1 SQL Server 2000 的版本和 组件	162
8.1.2 服务管理器	163
8.1.3 企业管理器	163
8.1.4 osql 实用工具	166
8.1.5 SQL 查询分析器	167
8.2 创建和维护数据库	168
8.2.1 数据库的概念及组成	168
8.2.2 创建数据库	169
8.2.3 删除数据库	171
8.2.4 备份数据库	171
8.2.5 还原数据库	173
8.2.6 分离数据库	174
8.2.7 附加数据库	175
8.3 表管理	176
8.3.1 表的概念	176
8.3.2 创建表	177
8.3.3 修改表	185
8.3.4 删除表	186

7.3.1 UpdatePanel 控件基础	150
7.3.2 使用 ContentTemplate 属性	151
7.3.3 ContentTemplateContainer 属性	152
7.3.4 Triggers 属性	153
7.3.5 在 ScriptManager 控件中注册 异步提交控件	154
7.4 使用 UpdateProgress 控件	155
7.4.1 UpdateProgress 控件基础应用	156
7.4.2 在页面中使用多个 UpdateProgress 控件	157
7.5 使用 Timer 控件	158
习题	160

8.3.5 查看和编辑表数据	187
8.4 表约束	190
8.4.1 主键约束	190
8.4.2 唯一性约束	191
8.4.3 检查约束	191
8.4.4 默认约束	192
8.4.5 外键约束	193
8.5 索引操作	195
8.5.1 设计索引	195
8.5.2 创建索引	195
8.5.3 查看、修改和删除索引	197
8.6 视图管理	197
8.6.1 创建视图	197
8.6.2 修改视图	202
8.6.3 删除视图	203
习题	203

第 9 章 常用 SQL 语句

9.1 SELECT 语句	205
9.1.1 本章实验的演示数据	205
9.1.2 SELECT 语句的基本 语法	206
9.1.3 指定要查询的列	207
9.1.4 显示唯一数据	207

9.1.5	显示列标题	208
9.1.6	设置查询条件	208
9.1.7	对结果集进行排序	210
9.1.8	使用 TOP n [PERCENT]	212
9.1.9	使用统计函数	213
9.1.10	使用分组统计	213
9.1.11	生成汇总行	215
9.1.12	连接查询	216
9.1.13	子查询	221
9.1.14	使用 UNION 关键字的合并查询	223
9.1.15	使用 CAST 和 CONVERT 函数进行类型转换	225
9.1.16	保存查询结果	226
9.2	数据更新语句	227
9.2.1	INSERT 语句	227
9.2.2	UPDATE 语句	230
9.2.3	DELETE 语句	232
	习题	233

第 3 部分 ASP.NET 应用实例开发

第 11 章 常用 ASP.NET 应用

	实例	268
11.1	实现用户登录和身份认证	268
11.1.1	登录控件综述	268
11.1.2	Membership 类	269
11.1.3	保存自定义用户信息的数据库结构设计	272
11.1.4	设计数据库访问类	273
11.1.5	使用 Login 控件设计登录用户页面	276
11.1.6	配置 Web.config 中的安全属性	279
11.1.7	在母版页中使用 Login 控件	281
11.1.8	使用 LoginStatus 控件和 LoginName 控件	284
11.1.9	注册新用户	286
11.1.10	使用 ChangePassword 控件修改用户密码	290

第 10 章 ADO.NET 数据库访问技术

10.1	ADO.NET 的结构和命名空间	234
10.1.1	ADO.NET 的结构	234
10.1.2	ADO.NET 的命名空间	235
10.2	ADO.NET 常用对象	236
10.2.1	Connection 对象	236
10.2.2	Command 对象	238
10.2.3	DataReader 对象	240
10.2.4	DataSet 对象	241
10.2.5	DataAdapter 对象	244
10.2.6	DataView 对象	245
10.3	服务器端数据访问控件	247
10.3.1	Repeater 控件	247
10.3.2	DataList 控件	250
10.3.3	GridView 控件	251
10.3.4	DetailsView 控件	260
	习题	264

11.1.11	使用 PasswordRecovery 控件找回密码	291
11.1.12	使用 LoginView 控件为不同用户显示不同的内容	294
11.2	几个常用的小实例	295
11.2.1	上传文件	295
11.2.2	发送电子邮件	297
11.2.3	实现图文验证功能	299

第 12 章 在线教育管理系统

12.1	系统分析与设计	303
12.1.1	系统功能描述	303
12.1.2	功能模块划分	304
12.1.3	系统流程分析	304
12.2	数据库结构设计与实现	305
12.2.1	创建数据库	305
12.2.2	数据库逻辑结构设计	305
12.3	目录结构与准备工作	308

12.3.1	设置 IIS 服务	308
12.3.2	目录结构	310
12.3.3	类文件	310
12.4	Admin 用户管理模块设计	314
12.4.1	登录主界面	314
12.4.2	设计主界面	315
12.4.3	设计 admin\index.aspx	316
12.4.4	设计 admin\left.aspx	316
12.5	学员管理模块设计	317
12.5.1	设计学籍管理页面	317
12.5.2	查看学员信息	319
12.5.3	设计交费管理页面	320
12.5.4	审核交费信息	323
12.6	业务统计模块设计	324
12.6.1	设计费用统计页面	324
12.6.2	设计人数统计页面	324
12.7	信息维护模块设计	325
12.7.1	设计课程分类页面	325
12.7.2	设计教师管理页面	331
12.8	课程管理模块设计	333
12.8.1	设计未发布课程页面	333
12.8.2	设计已发布课程页面	338
12.9	课件维护模块设计	338
12.9.1	设计课件管理页面	338
12.9.2	设计下载统计管理页面	341
12.10	系统主界面与登录程序设计	343
12.10.1	设计主界面	343
12.10.2	设计 index.aspx	344
12.10.3	设计 top.aspx	344
12.10.4	设计 left.aspx	344
12.10.5	设计 Default.aspx	345
12.10.6	注册用户登录程序设计	347
12.10.7	设计分类查看课程页面	347
12.10.8	设计查看课件页面	348
12.11	个人用户管理模块设计	350
12.11.1	注册新用户	350
12.11.2	个人用户管理界面设计	352
12.11.3	我的课程信息	352
12.11.4	我的课件信息	355
12.11.5	编辑个人信息	357

第 13 章 网上书城系统 359

13.1	系统分析与设计	359
13.1.1	系统功能描述	359
13.1.2	功能模块划分	360
13.1.3	系统流程分析	360
13.2	数据库结构设计与实现	361
13.2.1	创建数据库	361
13.2.2	数据库表设计	361
13.3	目录结构与数据库访问类	363
13.3.1	目录结构	363
13.3.2	数据库访问类	363
13.4	Admin 用户管理模块设计	367
13.4.1	登录主界面	367
13.4.2	设计主界面	368
13.4.3	设计 admin\index.aspx	368
13.4.4	设计 admin\left.aspx	369
13.5	图书分类管理模块设计	370
13.5.1	设计图书分类管理页面	370
13.5.2	添加图书分类信息	371
13.5.3	修改图书分类信息	372
13.5.4	删除图书分类信息	373
13.6	图书管理模块设计	374
13.6.1	显示图书列表	374
13.6.2	添加图书记录	375
13.6.3	修改图书信息	375
13.6.4	删除图书信息	376
13.7	订单管理模块设计	376
13.7.1	查看订单信息	377
13.7.2	订单处理	377
13.8	系统主界面与登录程序设计	378
13.8.1	设计主界面	378
13.8.2	注册用户登录	379
13.9	设计图书查询及购买模块	380
13.9.1	按分类查看图书列表	380
13.9.2	查看图书信息	381
13.9.3	选择要购买的图书	381
13.9.4	购物车管理	382
13.9.5	收银台管理	382
13.9.6	我的订单	383

13.10 用户管理设计	384
13.10.1 系统用户管理	384
13.10.2 注册新用户	384

第 4 部分

附录 1 实验

实验 1 ASP.NET 应用环境配置	388
实验 2 C#程序设计	390
实验 3 使用 ASP.NET 常用服务器控件	393
实验 4 使用 ASP.NET 内置对象	397
实验 5 ASP.NET 状态管理和应用程序配置	400
实验 6 ASP.NET AJAX 服务器端编程	403
实验 7 SQL Server 2000 数据库对象管理	405
实验 8 使用 ASP.NET 访问 SQL Server 数据库	409
大作业 网上同学录系统	411
项目 1 数据库结构设计	412

13.10.3 会员用户管理	385
13.10.4 查看用户明细信息	385
13.10.5 查看用户交易记录	386

附 录

项目 2 目录结构与通用模块	414
项目 3 系统主界面与登录程序设计	416
项目 4 管理主界面与登录程序设计	419
项目 5 个人信息管理	420
项目 6 公告信息管理模块设计	422
项目 7 照片分类管理模块设计	423
项目 8 照片管理模块设计	424
项目 9 留言板模块设计	425

附录 2 下载 SQL Server 2000

评估版	427
-----	-----

附录 3 SQL Server 2000 服务器

安装过程	429
------	-----

第1章

Web应用开发技术基础

第1部分

ASP.NET 程序设计基础

1.1 Web应用开发语言

在Web应用出现之前，“客户端服务器”（C/S）是应用系统的主流架构。每个应用系统都需要有一个客户端程序，它为用户提供操作界面，而数据通常存放在服务器端。在C/S架构的应用程序中，如果将服务器程序进行开发，通常需要同步开发客户端程序，只有双方的版本匹配，才能保证应用系统的正常运行。这大大增加了开发成本，影响了产品的推广。

Web应用的出现解决了上述问题。Web应用系统使用Web文档（网页）来表现用户界面，而Web文档都遵循标准HTML格式（包括2000年推出的XML标记语言）。因此，只要有Web浏览器就能浏览Web文档，所以在客户端可以使用不同类型的Web浏览器查看网页内容。只要用户按照安装一种Web浏览器，就可以查看所有Web文档，从而解决了不同应用系统兼容不同客户端程序的问题。

Web应用系统的部署放置在服务器端，客户端只接收经过服务器端处理的静态网页，因此也不存在客户端与服务器端兼容的问题。

1990年，在欧洲共同体的一个大型科研项目担任负责人的英国人Tim Berners-Lee发明了WWW（World Wide Web），通过Web，用户可以访问一个网页里比较直接地表示出互联网上的资源。最初，Web页面都是静态的，用户可以通过单击超链接来访问服务器上的不同网页。早期的Web服务器只能简单地响应浏览器发来的HTTP请求，并将存储在服务器上的HTML文件返回给浏览器。最早能够动态生成HTML页面的技术是CGI（Common Gateway Interface）。1993年，CGI 1.0的标准草案由NCSA（National Center for Supercomputing Applications）提出，1995年，NCSA开始制定CGI 1.1标准，1997年，CGI 1.2也颁布了以事日程。CGI技术允许服务器端的程序根据客户端的请求，动态生成HTML页面，这使客户端和服务端的动态信息交换成为可能。早期的CGI程序大多是编译后的可执行程序，其编程语言可以是C、C++、Pascal等任何通用的程序设计语言，也可以是Perl、Python等脚本语言。

1994年，Rasmus Lerdorf发明了用于Web服务器端编程的PHP（Personal Home Page Tools）语言。与以往的CGI程序不同，PHP语言将HTML代码和PHP指令结合成为完整的服务器端动态

第 1 章

Web 应用程序设计与开发概述

随着互联网技术的应用和普及,人类社会已经进入了信息化的网络时代,开发 Web 应用程序已经成为程序员的基本技能。本章首先介绍 Web 应用程序设计语言的基本情况,使读者从宏观上了解开发 Web 应用程序需要掌握哪些技术,为学习本书后面的内容奠定基础。

1.1 Web 应用程序设计语言的产生与发展

在 Web 应用程序出现之前,“客户端/服务器”(C/S)是应用程序的主流架构。每个应用程序都需要有一个客户端程序,它为用户提供管理和操作界面,而数据通常保存在服务器端。在 C/S 架构的应用程序中,如果对服务器端程序进行升级,通常需要同时升级客户端程序,只有双方的版本匹配,才能保证应用程序的正常运行。这无疑增加了维护成本,影响了产品的推广。

Web 应用程序则解决了上述问题。Web 应用程序使用 Web 文档(网页)来表现用户界面,而且 Web 文档都遵循标准 HTML 格式(包括 2000 年推出的 XHTML 标准格式)。因为所有 Web 文档都遵循标准化的格式,所以在客户端可以使用不同类型的 Web 浏览器查看网页内容。只要用户选择安装一种 Web 浏览器,就可以查看所有 Web 文档,从而解决了为不同应用程序安装不同客户端程序的问题。

Web 应用程序的代码都放置在服务器端,客户端只接收经过服务器端处理的静态网页,因此也不存在客户端与服务器端程序不匹配的问题。

1990 年,在欧洲共同体的一个大型科研机构任职的英国人 Tim Berners-Lee 发明了 WWW (World Wide Web)。通过 Web,用户可以在一个网页里比较直观地表示出互联网上的资源。最初的 Web 页面都是静态的,用户可以通过单击超链接等方式与服务器进行交互,访问不同的网页。

早期的 Web 服务器只能简单地响应浏览器发送过来的 HTTP 请求,并将存储在服务器上的 HTML 文件返回给浏览器。最早能够动态生成 HTML 页面的技术是 CGI (Common Gateway Interface)。1993 年,CGI 1.0 的标准草案由 NCSA (National Center for Supercomputing Applications)提出,1995 年,NCSA 开始制定 CGI 1.1 标准,1997 年,CGI 1.2 也被纳入了议事日程。CGI 技术允许服务端的应用程序根据客户端的请求,动态生成 HTML 页面,这使客户端和服务端的动态信息交换成为了可能。早期的 CGI 程序大多是编译后的可执行程序,其编程语言可以是 C、C++、Pascal 等任何通用的程序设计语言,也可以是 Perl、Python 等脚本语言。

1994 年,Rasmus Lerdorf 发明了专门用于 Web 服务端编程的 PHP (Personal Home Page Tools) 语言。与以往的 CGI 程序不同,PHP 语言将 HTML 代码和 PHP 指令结合成为完整的服务端动态

页面,程序员可以用一种更加简便、快捷的方式实现动态 Web 功能。

1995 年, Netscape 公司推出了一种在客户端运行的脚本语言, 即 JavaScript。使用 JavaScript 语言可以在客户端的用户界面上添加一些动态的元素, 如弹出一个对话框。

1996 年, Macromedia 公司推出了 Flash, 一种矢量动画播放器。它可以作为插件添加到浏览器中, 从而在网页中显示动画。

同样在 1996 年, Microsoft 公司推出了 ASP 1.0。这是 Microsoft 公司推出的第 1 个服务器端脚本语言, 使用 ASP 可以生成动态的、交互式的网页。从 Windows NT 4.0 开始, 所有的 Windows 服务器产品都提供 IIS (Internet Information Services) 组件, 它可以提供对 ASP 语言的支持。在 ASP 中, 可以使用 VBScript、JavaScript 等脚本语言开发服务器端 Web 应用程序。

1997~1998 年, Servlet 技术和 JSP 技术相继问世, 这两者的组合 (还可以加上 JavaBean 技术) 让 Java 开发者同时拥有了类似 CGI 程序的集中处理功能和类似 PHP 的 HTML 嵌入功能。此外, Java 的运行时编译技术也大大提高了 Servlet 和 JSP 的执行效率。

Sun 公司的 J2EE 是纯粹基于 Java 的解决方案。到 2003 年, Sun 的 J2EE 版本已经升级到了 1.4 版, 其中 3 个关键组件的版本也演进到了 Servlet 2.4、JSP 2.0 和 EJB 2.1。

2002 年, Microsoft 公司正式发布 .NET Framework 和 Visual Studio.NET 开发环境。它引入了 ASP.NET 这样一种全新的 Web 开发技术。ASP.NET 可以使用 VB.NET、C# 等编译型语言, 支持 Web Form、.NET Server Control、ADO.NET 等高级特性。

本节概要地介绍了 Web 应用程序产生和发展过程中一些主要技术的推出和应用情况, 使读者对 Web 应用程序的历史形成宏观的认识。

1.2 Web 应用程序的基本开发流程和工作原理

Web 应用程序通常由 HTML 文件、脚本文件和一些资源文件组成。

- HTML 文件可以提供静态的网页内容, 这也是早期最常用的网页文件。
- 脚本文件可以提供动态网页。ASP 的脚本文件扩展名为 .asp, PHP 的脚本文件扩展名为 .php, JSP 的脚本文件扩展名为 .jsp, ASP.NET 的脚本文件扩展名为 .aspx。
- 资源文件可以是图片文件、多媒体文件、配置文件等。

在完成需求分析和总体设计的情况下, 开发 Web 应用程序的基本流程如图 1.1 所示。

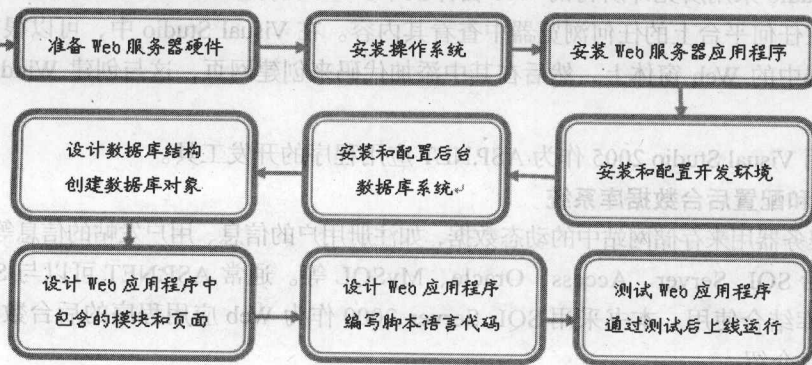


图 1.1 开发 Web 应用程序的基本流程

1. 准备 Web 服务器

运行 Web 应用程序需要一个载体,即 Web 服务器。一个 Web 服务器可以放置多个 Web 应用程序,也可以把 Web 服务器称为 Web 站点。

通常服务器有两层含义,一方面它代表计算机硬件设备,用来安装操作系统和其他应用软件;另一方面它又代表安装在硬件服务器上的相关软件。

要配置 Web 应用程序,首先需要准备一台硬件服务器,如果没有特殊需要,选择普通的 PC 服务器即可。PC 服务器的组件与普通计算机相似,主要包括主板、CPU、内存、硬盘、显卡等。只是 PC 服务器比普通计算机拥有更高的性能和更好的稳定性。在开发和测试阶段,或者比较小的网络环境下,也可以使用普通计算机作为 Web 服务器。

2. 安装操作系统

操作系统是控制其他程序运行、管理系统资源,并为用户提供操作界面的系统软件的集合。准备硬件 Web 服务器后,需要安装适当的操作系统。如果选择 ASP.NET 作为开发 Web 应用程序的语言,则需要安装 Windows Server 操作系统,如 Windows Server 2000、Windows Server 2003、Windows Server 2008 等。如果选择 PHP 和 Java 等支持跨平台的开发语言,也可以安装 UNIX 或 Linux 操作系统。

3. 安装 Web 服务器应用程序

Web 服务器应用程序可以响应用户通过浏览器提交的请求。如果用户请求执行的是 ASP.NET 脚本,则 Web 服务器应用程序将执行 ASP.NET 代码,最后将结果转换成 HTML 格式,并返回到前台,显示在浏览器中。

常用的 Web 服务器应用程序包括 IIS 和 Apache 等。选择 ASP.NET 作为 Web 应用程序的开发语言时,通常选择 IIS 作为 Web 服务器应用程序,因为它们都是微软的产品。本书将在 2.1 节介绍 IIS 的安装、配置和管理。

4. 安装和配置脚本语言编辑工具

作为 ASP.NET 的前身,ASP 没有提供一个集成的开发环境,也没有专用的编辑工具。可以使用任何文本编辑工具编辑 ASP 文本,包括 Windows 记事本。在这方面,ASP.NET 做出了显著的改进。可以在功能强大的 Visual Studio 中使用 VB.NET 或 C#等多种语言开发 ASP.NET 应用程序。

Visual Studio 是一套完整的开发工具集,用于生成 ASP.NET Web 应用程序、桌面应用程序和移动应用程序等。在 Visual Studio 中,Visual Basic、Visual C++、Visual C#和 Visual J#等语言都使用相同的集成开发环境(IDE),利用此 IDE 可以共享工具且有助于创建混合语言解决方案。

Visual Studio 采用所见即所得的 Web 窗体技术。Web 窗体呈现为浏览器兼容的 HTML 网页和脚本,可以在任何平台上的任何浏览器中查看其内容。在 Visual Studio 中,可以很方便地将控件拖放到设计器中的 Web 窗体上,然后在其中添加代码来创建网页,这与创建 Windows 窗体的方法相似。

本书采用 Visual Studio 2005 作为 ASP.NET 应用程序的开发工具。

5. 安装和配置后台数据库系统

数据库服务器用来存储网站中的动态数据,如注册用户的信息、用户发帖的信息等。常用的数据库服务器包括 SQL Server、Access、Oracle、MySQL 等。通常 ASP.NET 可以与 SQL Server 和 Access 数据库结合使用。本书采用 SQL Server 2000 作为 Web 应用程序的后台数据库,相关内容将在第 8 章介绍。

步骤 3~5 可以按照任意顺序执行。

6. 设计数据库结构, 创建数据库对象

在完成需求分析和总体设计后, 程序员需要根据总体设计的要求设计具体的数据库结构, 包括创建数据库、决定数据库中包含哪些表和视图、设计表、视图结构等。

在设计数据库结构后, 可以通过编写数据库脚本来创建这些数据库对象, 也可以手动创建数据库对象, 然后备份数据库, 以便在其他数据库服务器上还原数据库。

7. 设计 Web 应用程序, 编写代码

这是本书要介绍的重点内容: 使用 ASP.NET 设计网页代码, 完成 Web 应用程序的具体功能。

通常程序员需要根据总体设计文档将每个功能模块划分成若干个网页文件; 然后使用 Dreamweaver 设计网页的基本框架和网页中的静态元素 (如表格、静态图像、静态文本等); 最后在网页中添加 ASP.NET (Visual#) 代码, 完成网页的具体功能。

8. 测试 Web 应用程序, 通过测试后上线运行

在 Web 应用程序开发完成后, 需要设计测试案例, 测试其具体功能的实现情况。在通过测试达到实际应用的需求后, 可以将 Web 应用程序部署到 Web 服务器上。通常需要准备一个备份 Web 服务器, 以便实现数据备份, 并且在增加新功能时提供测试环境。

Web 应用程序的工作流程如图 1.2 所示。

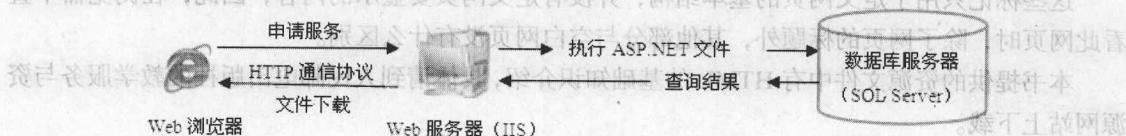


图 1.2 Web 应用程序的工作流程

1.3 当前比较流行的 Web 应用程序设计语言

在 1.1 节中提到了一些 Web 应用程序设计语言, 本节将对其中比较流行的语言进行简单的介绍。

1.3.1 HTML 简介

HTML 是英文 HyperText Markup Language 的缩写, 中文的意思是超文本标记语言。它是通过嵌入代码或标记的方式来表明文本格式的国际标准。用它编写的文件扩展名是 .html 或 .htm, 这种网页文件的内容是静态的, 而且无法与后台数据库结合使用。

虽然可以使用很多可视化工具设计网页, 但是在设计 ASP.NET 脚本时, 经常需要在代码中直接使用到 HTML, 以实现不同的网页效果。

HTML 中包含很多 HTML 标记, 它们可以被 Web 浏览器解释, 从而决定网页的结构和显示的内容。这些标记通常成对出现, 如 <HTML> 和 </HTML> 就是常用的标记对, 语法格式如下:

<标记名> 数据 </标记名>

本小节将介绍一些基本结构标记。HTML 文档可以分为两部分, 即文件头与文件体。文件头中提供了文档标题, 并建立 HTML 文档与文件目录间的关系; 文件体部分是 Web 页的实质内容, 它是 HTML 文档中最主要的部分, 其中定义了 Web 页的显示内容和效果。

常用的结构标记如表 1.1 所示。

表 1.1

HTML 常用的结构标记

结构标记	具体描述
<HTML>...</HTML>	标记 HTML 文档的开始和结束
<HEAD>...</HEAD>	标记文件头的开始和结束
<TITLE>...</TITLE>	标记文件头中的文档标题
<BODY>...</BODY>	标记文件体部分的开始和结束
<!--...-->	标记文档中的注释部分

这些基本结构标记文档的使用实例如下所示：

```
<HTML>
<HEAD>
  <TITLE> HTML 文件标题.</TITLE>
</HEAD>
<BODY>
  <!-- HTML 文件内容 -->
</BODY>
</HTML>
```

这些标记只用于定义网页的基本结构，并没有定义网页要显示的内容，因此，在浏览器中查看此网页时，除了网页的标题外，其他部分与空白网页没有什么区别。

本书提供的资源文件中有 HTML 的基础知识介绍，具体请到人民邮电出版社的教学服务与资源网站上下载。

1.3.2 ASP 简介

ASP 是 Active Server Pages 的缩写，是 Microsoft 公司开发的 Web 服务器端脚本开发环境，利用它可以生成动态、高效的 Web 应用程序。

虽然人们习惯于将 ASP 称为 ASP 语言，但从严格意义上讲，ASP 只是为 VBScript、Java Script 等脚本语言提供了一个运行的环境，使开发人员可以在 HTML 代码中使用脚本语言编写程序。当然，ASP 自身也提供了一些非常好用的命令和内置对象。

ASP 程序保存为后缀名为 .asp 的文件，一个 ASP 文件相当于一个可执行文件，因此，必须放在 Web 服务器上有可执行权限的目录下。当浏览器向 Web 服务器请求调用 ASP 文件时，就启动了 ASP。Web 服务器开始调用 ASP，完整读入被请求的 .asp 文件，执行每一个命令，然后动态生成一个 HTML 页面并送到浏览器。由于 ASP 在服务器端解释执行，开发者可以不必考虑浏览器是否支持 ASP，也不必担心程序会被从客户端下载。

1.3.3 JSP 简介

JSP 是 Java Server Pages 的缩写，是由 Sun 公司主导推出的一种动态网页技术标准。JSP 与 ASP 类似，它是在传统的网页 HTML 文件中插入 Java 程序段和 JSP 标记，从而形成 JSP 文件。JSP 文件的扩展名为 .jsp。

使用 JSP 开发的 Web 应用程序是跨平台的，既能在 Linux 环境下运行，也能在 Windows 等其他操作系统上运行。JSP 将网页逻辑与网页设计和显示分离，支持可重用的基于组件的设计，使开发 Web 应用程序变得迅速和容易。

Web 服务器在遇到访问 JSP 网页的请求时，首先执行其中的程序段，然后将执行结果连

同 JSP 文件中的 HTML 代码一起返回给客户。插入的 Java 程序段可以操作数据库、重新定向网页等,以实现建立动态网页所需要的功能。

1.3.4 PHP 简介

PHP 是服务器端、跨平台、HTML 嵌入式的脚本语言。与 ASP 语言一样,使用 PHP 可以开发基于 Web 的网络应用程序。

PHP 脚本文件的扩展名为.php,其中包含 HTML 代码和 PHP 代码。Apache 服务器在接收到 PHP 脚本文件的请求后,会解析 PHP 脚本文件中的 PHP 代码,执行代码并将其转换为 HTML 格式,然后转送到客户端。

1.3.5 ASP.NET 简介

ASP.NET 并不是 ASP 的简单升级,而是基于 .NET Framework 技术的新一代 Web 应用程序开发系统,可以在 Web 服务器上建立的功能强大的 Web 应用程序。

ASP.NET 与 ASP 的主要区别如下。

1. 开发语言不同

ASP 只允许使用 VBScript 和 JavaScript 等脚本语言,功能相对简单;而 ASP.NET 允许用户选择功能强大的 Visual Basic.NET 和 Visual C# 等高级开发语言,还可以使用 .NET Framework 提供的丰富的类库。

2. 运行机制不同

ASP 是在执行时对脚本进行解释,因此其执行效率比较低。ASP.NET 和其他高级程序设计语言一样,采用编译式的编程框架。ASP.NET 代码事先编译成功后,再部署到 Web 服务器上。当执行代码时,运行的是 Web 服务器上已经编译好的公共语言运行时库代码,因此可以提高效率。

3. 开发方式不同

在使用 ASP 设计网页时,界面设计和程序设计混在一起,程序的结构比较乱,维护进来比较困难;ASP.NET 把界面设计和程序设计分离,分别保存在不同的文件中,程序结构清晰,便于维护。

习 题

一、选择题

- 下面不属于 Web 应用程序组成部分的是 ()。
A. HTML B. 脚本文件 C. Web 站点 D. 资源文件
- 下面说明正确的是 ()。
A. 只能使用 DreamWeaver 编辑 ASP.NET 代码
B. 只能使用 C# 语句开发 ASP.NET 应用程序
C. ASP.NET 是基于 .NET Framework 技术的新一代 Web 应用程序开发系统
D. 只能使用 SQL Server 作为 ASP.NET 的后台数据库
- 下面能在 Linux 下运行的 Web 应用程序开发语言是 ()。
A. ASP B. ASP.NET C. JSP D. C#