

高等院校计算机教育系列教材



ASP.NET Web 开发教程

程不功 编著

- 内容简明扼要，突出知识要点。
- 知识点新，突出实践教学，强化能力培养。
- 深入浅出地阐述设计思想和相关理论。
- 本书内容涵盖以ASP.NET 3.5 Web开发的各项关键技术(C#语言)。

清华大学出版社

高等院校计算机教育系列教材

ASP.NET Web 开发教程

程不功 编著

清华大学出版社

北 京

内 容 简 介

这是一本 ASP.NET 3.5 Web 开发的教程(C#版)。全书共分 7 部分,包括 29 章和两个附录。内容涵盖 ASP.NET 基础、浏览器端开发、服务器端开发、母版页与角色管理、ASP.NET Ajax 技术、XML Web 服务以及综合示例等。

本书的特点是,以 Web 应用开发为主线组织内容;通过大量实例深入浅出地讲解技术,引导读者利用 ASP.NET 3.5 快速开发出功能强大、运行可靠而且易于扩展的系统。

本书的主要对象是计算机应用、信息专业的大专院校学生,初、中级程序员或者 Web 的业余爱好者。对于一些想要自行开发 Web 的自学者,本书也是一本很好的自学教材。

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

ASP.NET Web 开发教程/程不功编著. —北京:清华大学出版社, 2011.9

(高等院校计算机教育系列教材)

ISBN 978-7-302-26722-5

I. ①A… II. ①程… III. ①网页制作工具—程序设计—教材 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2011)第 177025 号

责任编辑:汤涌涛

装帧设计:杨玉兰

责任校对:李玉萍

责任印制:何 芊

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:010-62776969, c-service@tup.tsinghua.edu.cn

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

印 刷 者:北京嘉实印刷有限公司

装 订 者:三河市漂源装订厂

经 销:全国新华书店

开 本:185×260 印 张:29.5 字 数:714 千字

版 次:2011 年 9 月第 1 版 印 次:2011 年 9 月第 1 次印刷

印 数:1~4000

定 价:49.00 元

产品编号:042908-01

前 言

微软 2008 年推出的 ASP.NET 3.5 是在 ASP.NET 2.0 的基础上发展起来的，一方面它进一步完善了 2.0 版本，另一方面又增添了很多最新技术，但仍然与 2.0 版本兼容，甚至允许两种版本安装在同一台机器上。而且微软公司已经公开承诺允许长期免费使用这两个系统。

本书的特点是，以 Web 应用开发为主线组织内容；通过大量实例深入浅出地讲解技术，引导读者利用 ASP.NET 3.5 快速开发出功能强大、运行可靠而且易于扩展的系统。

本书不是一本“傻瓜书”，也不是一本“应用手册”，而是一本“教程”。就是说，书中不仅要说明技术“是什么”，还要说明技术产生的历史背景、解决的思路和方法，以及相关理论。

学习本书之前最好已经学过一门程序设计语言(C、C++、Java、VB 等都可以)，如果学过 C#.NET 当然更好，没有学过 C#.NET 也没有多大关系，必要时可参阅本书的附录。

本书共分七大部分，包括 29 章和两个附录。

第一部分：ASP.NET 基础。介绍.NET 框架的组成、作用以及 ASP.NET 网站的逻辑结构。

第二部分：浏览器端开发。浏览器端的开发技术近几年来有很大发展，网页质量已经成为网站竞争的重要因素。这一部分讲授 HTML、CSS、动态 HTML(即 DHTML)以及利用 jQuery 设计动画等四方面的内容。

第三部分：服务器端开发。这是本书的重点。ASP.NET Web 应用程序(网站)就是一种基于服务器的系统，它包括的内容很多，因此用 12 章的篇幅进行讲解，主要讲授 ASPX 网页以及各种服务器控件的使用方法；各种数据访问技术，包括 ADO.NET 的系统结构，对数据库的连接、显示、编辑与同步，以及存储过程、数据缓存的方法；最后介绍 LINQ 这项最新技术及其应用等。

第四部分：母版页与角色管理。网站安全、网站显示风格等都是网站开发中必须注意的主题。在这部分中将站在全局的高度来讲述网站的显示风格、网站导航、网站安全以及网站的个性化服务等四方面的内容。

第五部分：ASP.NET Ajax 技术。用 3 章篇幅主要讲授 Ajax 原理、ASP.NET Ajax 技术以及 Ajax ToolKit 的使用等三方面的内容，使读者在学会原理的基础上，能够对传统的 Web 网站进行 Ajax 功能升级。

第六部分：Web 服务。主要通过几个典型实例让读者掌握 XML Web 服务的调用方法，以及创建 XML Web 服务网站两方面的问题。

第七部分：综合示例。在这里选择了网上招聘及留言板、快速开发动态数据驱动网站、创建电子商务网站三个典型项目，通过讲述它们的设计方法，使读者进一步掌握利用前面所学内容综合开发系统的实际本领。

本书是《ASP.NET 2.0 动态网站开发教程》的姊妹篇。2006 年上半年本书的作者曾经出版了《ASP.NET 2.0 动态网站开发教程》，2008 年又出版了它的第 2 版，这两本书受到

了不少读者的喜爱(4年中已经连续印刷了14次),同时也要感谢很多读者提出的批评和建议,现在这本书保留了前两本书的撰写风格和部分内容,因此可以说本书既站在 Web 发展的前沿又保持了原书的优点。

本书的主要读者对象是计算机应用、信息专业的大专院校学生,初、中级程序员或者 Web 的业余爱好者。

编 者

目 录

第一部分 ASP.NET 基础

第 1 章 .NET 框架简介	2	1.7 习题	12
1.1 Web 发展历史的简要回顾	2	第 2 章 ASP.NET 网站的体系结构	13
1.1.1 从静态网页发展到动态网页	2	2.1 ASP.NET 网站的逻辑结构	13
1.1.2 动态网页发展的几个阶段	3	2.2 ASP.NET 网站的组成	14
1.2 ASP.NET 发展的简要历史	4	2.2.1 虚拟目录	14
1.3 .NET 框架(Framework)的发展过程	4	2.2.2 网页文件	14
1.3.1 什么是.NET 框架	4	2.2.3 网站配置文件	15
1.3.2 .NET 框架的发展过程	4	2.2.4 网站全局文件	16
1.4 .NET 框架的基础结构	5	2.2.5 几个专用的共享目录	16
1.4.1 .NET 框架中的程序开发语言	6	2.3 创建新网站	17
1.4.2 基础类库	7	2.3.1 文件系统网站	17
1.4.3 公共语言运行库(CLR)	7	2.3.2 本地 IIS 网站	17
1.5 XML: 可扩展的标记语言	8	2.3.3 远程网站	18
1.5.1 什么是 XML	8	2.4 小结	18
1.5.2 XML 的特点	10	2.5 习题	19
1.6 小结	11		

第二部分 浏览器端开发

第 3 章 HTML	22	3.4.1 html 文档编辑器的选择	26
3.1 HTML 概述	22	3.4.2 文档编辑的基本步骤	26
3.2 HTML 标记的基础	23	3.5 html 文本编辑	26
3.2.1 基本的 HTML 语法	23	3.5.1 文本格式化	26
3.2.2 标记的属性	23	3.5.2 列表	29
3.2.3 注释语句	24	3.5.3 表格和图层	30
3.3 html 文档的结构	24	3.6 插入图像	33
3.3.1 html 标记	24	3.6.1 图像的类型	33
3.3.2 首部标记	24	3.6.2 插入图像的方法	34
3.3.3 正文标记	25	3.6.3 通过属性编辑图像	34
3.3.4 html 文档的基本结构	25	3.7 超链接	35
3.4 html 文档的编辑工具	26	3.7.1 超链接的概念	35

3.7.2 常见链接的创建	36	5.4 DHTML 的应用示例	78
3.7.3 示例	37	5.5 小结	82
3.8 创建移动的文本文本	38	5.6 习题	82
3.9 HTML 与 XML 的比较	39	第 6 章 利用 jQuery 设计动画	84
3.10 HTML 表单及其控件	40	6.1 jQuery 基础	84
3.10.1 表单(form)的作用	40	6.1.1 什么是 jQuery	84
3.10.2 HTML 的表单控件	40	6.1.2 jQuery 能做什么	85
3.10.3 表单示例	42	6.1.3 JQuery 的特点	85
3.11 小结	43	6.1.4 配置 jQuery 的使用环境	85
3.12 习题	44	6.1.5 jQuery 的起始语句	86
第 4 章 CSS	46	6.1.6 示例	86
4.1 CSS 的基本概念	46	6.2 jQuery 对元素定位	87
4.1.1 什么是 CSS	46	6.2.1 定位集合对象	87
4.1.2 CSS 的作用	46	6.2.2 定位单个(或部分)对象	89
4.2 CSS 的定义方法	47	6.2.3 根据层次关系对元素定位	90
4.2.1 两种定义方式	47	6.2.4 在遍历 DOM 中进行定位	90
4.2.2 定义语句	47	6.3 事件与方法	91
4.3 CSS 网页布局	50	6.3.1 事件	91
4.3.1 概述	50	6.3.2 成对事件代码的简化	93
4.3.2 网页中的框架模型	50	6.3.3 事件冒泡	94
4.3.3 利用 ASP.NET 3.5 的 工具进行布局	54	6.4 对节点的操作	94
4.3.4 对内容溢出的处理	59	6.4.1 创建新节点	94
4.4 小结	60	6.4.2 删除节点	95
4.5 习题	60	6.4.3 复制节点	95
第 5 章 动态 HTML 技术	62	6.5 样式操作	96
5.1 动态 HTML 的基本理论	62	6.5.1 获取样式	96
5.1.1 DHTML 基本概念	62	6.5.2 增加样式	96
5.1.2 DOM	63	6.5.3 删除样式	96
5.2 JavaScript 语言	67	6.5.4 判断元素是否含有某样式	96
5.2.1 JavaScript 语言简介	67	6.6 利用 jQuery 设计动画	97
5.2.2 JavaScript 的基本用法	67	6.6.1 show()与 hide()方法	97
5.3 多媒体的引用	77	6.6.2 fadeIn()与 fadeOut()方法	97
5.3.1 内部多媒体文件的引用	77	6.6.3 slideUp()与 slideDown()方法	97
5.3.2 外部多媒体文件的引用	77	6.6.4 animate()方法	97
		6.6.5 示例	98

6.6.6 停止元素动画	99	6.8.2 简单插件的编写及使用	105
6.7 动画设计示例	99	6.8.3 使用 jQuery UI 插件的范例	106
6.8 插件(Plug in)	104	6.9 小结	110
6.8.1 概述	104	6.10 习题	110

第三部分 服务器端开发

第 7 章 ASPX 网页及代码的 存储模式

7.1 ASPX 网页的基类	114
7.2 ASPX 网页代码的存储模式	114
7.2.1 代码分离模式	115
7.2.2 代码的单文件模式	118
7.3 ASPX 网页中的表单	119
7.4 代码模式的选择	120
7.5 小结	120
7.6 习题	120

第 8 章 标准控件与事件模型

8.1 网页中的控件	122
8.1.1 控件类型	122
8.1.2 网页标准控件	122
8.2 ASP.NET 的事件处理模型	130
8.2.1 基于服务器的处理模型	130
8.2.2 尽量减少信息的往返次数	130
8.2.3 结合浏览器处理事件	131
8.3 应用示例	133
8.4 小结	139
8.5 习题	140

第 9 章 状态管理

9.1 状态的类型	142
9.2 视图状态	142
9.3 应用程序状态	144
9.4 会话状态	145
9.4.1 概述	145
9.4.2 Session 对象中方法的调用	146
9.5 Cookie 状态	147
9.6 简单的应用示例	149

9.7 Web 窗体页的生命周期	153
9.8 小结	154
9.9 习题	154

第 10 章 数据验证

10.1 概述	156
10.2 验证控件的类型	156
10.3 各验证控件的使用方法	157
10.3.1 RequiredFieldValidator 控件	157
10.3.2 CompareValidator 控件	158
10.3.3 RangeValidator 控件	158
10.3.4 RegularExpressionValidator 控件	160
10.3.5 ValidationSummary 控件	160
10.4 自定义控件	160
10.5 分组校验技术	161
10.6 拒绝机器人行为	163
10.6.1 概述	163
10.6.2 创建图形验证网页	164
10.6.3 与机器人斗争的长期性	169
10.7 综合示例	170
10.8 小结	170
10.9 习题	171

第 11 章 ADO.NET 简介

11.1 从 ODBC 到 ADO 数据库的 通用接口	173
11.1.1 ODBC 通用接口	173
11.1.2 ADO 通用接口	174

11.2 ADO.NET 的数据模型	174	13.2 数据表同步	204
11.2.1 数据访问的层次结构	175	13.2.1 概述	204
11.2.2 数据集与数据提供者	175	13.2.2 同一窗体页中父、 子表同步	205
11.3 数据源控件	179	13.2.3 不同窗体页中父、 子表同步	206
11.3.1 概述	179	13.3 合并多表显示	207
11.3.2 数据源控件的类型	179	13.4 小结	209
11.4 小结	181	13.5 习题	209
11.5 习题	181		
第 12 章 利用 GridView 控件		第 14 章 编辑数据表	211
显示数据	183	14.1 数据表编辑的 SQL 语句	211
12.1 数据绑定的基本概念	183	14.2 使用 GridView 控件更新数据表	212
12.2 SQL Server 2005(2008) Express Edition 简介	184	14.3 使用 GridView 控件的列模板	214
12.2.1 SQL Server 2005(2008) Express Edition 的主要特点	184	14.3.1 选择显示的字段	214
12.2.2 在网站中创建 Express Edition 数据库	185	14.3.2 增添按钮	215
12.3 连接数据库	188	14.3.3 使用模板列	215
12.4 对数据表进行分页、排序和选择	191	14.4 使用 GridView 控件增添记录	219
12.4.1 分页	192	14.5 使用 DetailsView 控件	220
12.4.2 排序	192	14.6 小结	223
12.4.3 选择	192	14.7 习题	223
12.5 利用模板美化显示	193	第 15 章 ListView 与 DataPager 控件	225
12.5.1 模板	193	15.1 ListView 控件中的模板	225
12.5.2 自动套用格式	194	15.2 模板中绑定数据的方法	226
12.5.3 设置模板样式	194	15.3 用网格方式显示数据	226
12.6 显示记录中的图像	195	15.3.1 设计步骤	226
12.7 小结	196	15.3.2 模板代码的分析	227
12.8 习题	196	15.3.3 修改后显示的界面	229
第 13 章 数据库查询与同步	198	15.4 用平铺方式显示数据	230
13.1 数据库查询	198	15.4.1 设计步骤	230
13.1.1 数据库查询语句	198	15.4.2 模板的代码分析	231
13.1.2 单一条件查询	199	15.5 小结	232
13.1.3 选择条件查询	201	15.6 习题	233
13.1.4 多条件的组合查询	202	第 16 章 存储过程与数据缓存	235
		16.1 概述	235

16.2 创建存储过程	236	第 18 章 LINQ 技术	257
16.2.1 在 SQL Server 2000 中创建 存储过程	236	18.1 概述	257
16.2.2 直接在应用程序的环境中 创建存储过程	237	18.2 LINQ 查询的语法基础	258
16.3 调用存储过程	238	18.3 Lambda 表达式	260
16.4 数据缓存	239	18.4 LINQ to SQL	263
16.4.1 网页输出缓存	239	18.4.1 将数据库映射成类和 对象	263
16.4.2 利用数据源控件缓存 数据库	240	18.4.2 映射中的对应关系	264
16.5 小结	242	18.4.3 映射后的部分代码	265
16.6 习题	242	18.4.4 数据库显示和查询	266
第 17 章 创建三层架构	244	18.5 利用 LINQ 编辑数据库	267
17.1 从两层架构发展成三层架构	244	18.5.1 更新数据表(Updating)	268
17.2 ASP.NET 3.5 中间层的特点	245	18.5.2 插入新记录(Inserting)	268
17.3 创建中间层的步骤	246	18.5.3 删除记录(Deleting)	269
17.4 在网页中调用中间层对象	248	18.6 使用 LINQ 数据源控件	269
17.4.1 直接调用中间层对象	248	18.7 调用存储过程	270
17.4.2 通过 ObjectDataSource 数据源控件调用中间层	249	18.8 利用 LINQ 分析数据	270
17.5 三层架构的应用示例	250	18.8.1 销售分析	271
17.6 小结	255	18.8.2 对产品销路的分析	271
17.7 习题	255	18.8.3 职工管理	271
		18.8.4 批量修改数据	272
		18.9 小结	273
		18.10 习题	273

第四部分 母版页与角色管理

第 19 章 主题、用户控件和母版页	276	19.2.2 创建用户控件的方法	280
19.1 主题	276	19.2.3 使用用户控件	281
19.1.1 什么是主题	276	19.2.4 代码分析	281
19.1.2 创建主题及皮肤文件的 方法	276	19.2.5 将 Web 窗体页转换为 用户控件	281
19.1.3 对同一控件多种定义的 方法	277	19.3 母版页	282
19.1.4 应用主题的方法	279	19.3.1 什么是母版页	282
19.2 用户控件	279	19.3.2 创建母版页的方法	282
19.2.1 什么是用户控件	279	19.3.3 在母版页中放入新网页的 方法	283

19.3.4 将已建成的网页放入母版页中	284	21.4.2 使用创建新用户控件	305
19.4 小结	285	21.4.3 登录状态与登录姓名控件	308
19.5 习题	285	21.4.4 登录视图控件	308
第 20 章 网站导航	287	21.4.5 PasswordRecovery 控件和 ChangePassword 控件	310
20.1 TreeView 控件	287	21.4.6 在 Login 控件中增添图片校验码	310
20.1.1 概述	287	21.5 直接调用 Membership API 方法	311
20.1.2 选择 TreeView 控件的视图	287	21.5.1 创建新客户	311
20.1.3 编辑节点	288	21.5.2 创建新角色	312
20.1.4 对节点事件的处理	289	21.5.3 给客户分配角色	313
20.2 站点地图文件	290	21.5.4 删除角色	314
20.3 将 TreeView 结合站点地图进行导航	291	21.5.5 从角色中删除客户	314
20.4 利用动态菜单进行导航	292	21.5.6 删除客户	315
20.4.1 结合站点地图创建动态菜单	292	21.6 小结	315
20.4.2 创建主菜单	292	21.7 习题	315
20.5 使用 SiteMapPath 控件	293	第 22 章 网站的个性化服务	317
20.6 小结	294	22.1 概述	317
20.7 习题	294	22.2 ASP.NET 3.5 对个性化设计的支持	318
第 21 章 基于角色的安全技术	296	22.2.1 关于 Membership	318
21.1 基于角色的安全技术特点	296	22.2.2 关于 profile	318
21.1.1 网站中可以包括多个入口	296	22.2.3 关于 WebPart	321
21.1.2 基于角色的安全技术是有层次的	297	22.3 保留客户关心的数据	321
21.2 ASP.NET 3.5 基于角色的安全技术特点	297	22.4 WebPart 介绍	323
21.3 基于角色安全技术的准备工作	298	22.4.1 定制网页时能够执行的任务	323
21.3.1 组织好站点中的文件	298	22.4.2 WebPart 的分类	324
21.3.2 利用网站管理工具进行安全配置	298	22.5 定制主页	324
21.4 利用控件创建安全网页	303	22.5.1 创建简单的包含 WebPart 控件的网页	324
21.4.1 客户登录控件	303	22.5.2 创建可以编辑和改变布局的网页	327
		22.5.3 运行中增添 WebPart 控件	331
		22.6 小结	332
		22.7 习题	332

第五部分 ASP.NET Ajax

第 23 章 Ajax 原理	336	24.2.1 客户端架构	350
23.1 概述	336	24.2.2 服务器端架构	351
23.1.1 传统的浏览器与服务器的 通过程程	336	24.3 ASP.NET Ajax 控件	351
23.1.2 Ajax 模式下的通过程程	336	24.4 服务器端 ASP.NET Ajax 应用示例	354
23.1.3 信息流通量的比较	337	24.5 给传统网页增添 Ajax 功能	355
23.2 Ajax 的组成	339	24.6 小结	356
23.3 Ajax 中的几个关键语句	340	24.7 习题	356
23.3.1 通信类的兼容语句	340	第 25 章 Ajax 工具箱	357
23.3.2 浏览器中元素定位语句	341	25.1 安装 ASP.NET Ajax 工具箱	357
23.3.3 异步通信的语句	341	25.2 设计 Accordion(可折叠面板)控件	358
23.4 Ajax 异步通信示例	341	25.2.1 Accordion 的嵌套结构	358
23.4.1 浏览器端的设置	341	25.2.2 Accordion 控件的 应用示例	360
23.4.2 服务器端的设置	343	25.3 几个支持 Button 的 Toolkit	361
23.5 JSON 语言	343	25.3.1 用于增加【确认】功能的 Toolkit	361
23.5.1 什么是 JSON	344	25.3.2 为控件增强立体感	362
23.5.2 数组字面量	344	25.4 使用几个支持 TextBox 的控件	362
23.5.3 对象字面量	344	25.4.1 对输入的数据类型 进行过滤	362
23.5.4 混合字面量	345	25.4.2 用按钮方式增减输入的 数字	363
23.5.5 JSON 语法	346	25.5 小结	363
23.5.6 JSON 与 XML 比较	346	25.6 习题	364
23.6 小结	348		
23.7 习题	348		
第 24 章 ASP.NET Ajax 技术	350		
24.1 ASP.NET Ajax 的特点	350		
24.2 ASP.NET Ajax 的架构	350		

第六部分 Web 服务

第 26 章 XML Web 服务	366	26.3.3 解释信息	368
26.1 XML Web 服务的特点	366	26.3.4 WS-*规范	368
26.2 XML Web 服务的过程	366	26.4 几个典型的应用	368
26.3 相关协议	367	26.4.1 调用气象预报服务	369
26.3.1 发现服务	367	26.4.2 调用国内航班信息	372
26.3.2 传输信息	368	26.4.3 调用股票信息	373

26.4.4 电视节目预报信息	375	26.6 小结	381
26.5 创建 XML Web 服务网站	378	26.7 习题	381
26.5.1 创建.aspx 文件	378		
26.5.2 创建温度转换的 Web 服务	378		

第七部分 综合示例

第 27 章 网上招聘与留言板	384	29.1 食品商店网站设计	400
27.1 FormView 控件简介	384	29.1.1 概述	400
27.2 利用 FormView 控件设计 招聘网页	384	29.1.2 主界面设计	403
27.3 利用 FormView 控件设计留言板	386	29.1.3 选择商品	406
27.3.1 打开留言板	387	29.1.4 创建购货车	406
27.3.2 留言网页的界面设计	387	29.1.5 结账	410
27.3.3 对留言板的管理	388	29.1.6 保存及显示订单	412
27.4 使用 Wizard 控件	388	29.1.7 放大图像介绍商品的 方法	414
27.4.1 Wizard 控件的用途	388	29.2 服装商店网站设计	418
27.4.2 Wizard 控件的结构	389	29.2.1 几张网页之间的联系	418
27.4.3 Wizard 控件的使用方法	389	29.2.2 主界面的设计	419
27.5 小结	390	29.2.3 跨页同步	420
27.6 习题	391	29.2.4 动态生成控件	420
第 28 章 快速创建动态数据 驱动网站	392	29.2.5 生成购货车	421
28.1 概述	392	29.3 账户管理	422
28.2 数据模型(Model)与 支架(Scaffold)	392	29.3.1 概述	422
28.3 创建步骤	393	29.3.2 准备工作	423
28.4 系统的目录结构	395	29.3.3 账户注册模块	424
28.5 系统的基本功能	395	29.3.4 账户验证模块	426
28.6 修改系统的方法	397	29.3.5 账户管理的信息流程	427
28.7 小结	398	29.4 小结	428
28.8 习题	398	29.5 习题	429
第 29 章 创建电子商务网站	400	附录 A C#.NET 基础语法参考	433
		附录 B 部分习题参考答案	448
		参考文献	458

第一部分

ASP.NET 基础

为了学会并掌握 ASP.NET 系统，需要对系统的发展过程、系统的整体概貌以及 .NET 框架的基础结构有一个初步的了解。这一部分讲述以下几方面的问题。

- Web 的发展简史。
- .NET 框架与 ASP.NET 发展的简要过程。
- .NET 框架的基础结构。
- ASP.NET 网站的逻辑结构与网站组成。
- 创建 ASP.NET 网站的方法。

第 1 章 .NET 框架简介

本章讲授的内容包括三部分。

- Web 以及 ASP.NET 发展历史的简要回顾。
- .NET 框架的基础结构。
- XML 语言。

后面两个问题是重点，在后续的讲授中将经常联系到这两方面的内容。

1.1 Web 发展历史的简要回顾

Web 的概念是 Tim Berners-Lee 于 1980 年在瑞士提出来的。1991 年 8 月出现了第一台 Web 服务器，到今天才不过 20 年的历史，Web 出现后的发展速度、影响的范围都远远超出了人们的预想。目前，它已经深入到人类社会，并广泛应用于人们的生活之中。

Web 通常由浏览器、服务器(通常还包括数据库)组成。其一部分是浏览器或其他数字设备，另一部分是服务器。最初的 Web 只能由服务器发送简单的单页静态纯文本，现在已经发展到支持音频、视频这样丰富内容的信息，还常常执行一些复杂的基于因特网的应用程序。在这些应用程序中，服务器与浏览器之间频繁地进行通信、交互以处理大量的数据，实际上已经成为一种分布式的应用系统。一个 Web 应用程序既是 Web 应用的基本单位，也是程序部署的基本单位，有时又称为网站。例如电子商务、电子政务、远程教育、网上银行、网上招聘、信息咨询等一些大大小小的网站都属于 Web 应用的范围。

1.1.1 从静态网页发展到动态网页

早期的 Web 服务器发送出的是静态网页，虽然网页中包括有文字和图片，但是只要不改变设计，网页的内容是不会变化的。对静态网页的访问过程如下。

- (1) 客户通过浏览器向服务器申请 URL 页面。
- (2) 服务器向客户送出被申请的页面。
- (3) 浏览器下载并解释传来的页面并显示在浏览器中。
- (4) 断开客户与服务器之间的联系。

整个过程比较简单，到浏览器下载完页面时为止，整个过程就结束了。用于发布静态网页的网站设计比较简单。这种设计对于早期的网站来说也许已足够。因为早期使用网站的大多是一些科学工作者，他们关注的重点只是交流有关科学技术的内容。

随着因特网应用领域的扩展，各种不同类型的客户加入到网络中来，不少客户很快就提出了新的要求。例如，有的客户提出，能不能代我查阅一下我银行存款的变化情况？要满足类似这样的需求，服务器的工作就不那么简单了。它首先要查阅银行账户，进行必要的计算和统计，再将结果反馈给客户。这就是说，服务器在回答问题前必须先执行一些相关的程序。这段程序不仅应能回答客户的问题，还要能够保障客户的信息安全，防止其他人进行查阅或破坏。

类似这种网页的输出内容将随程序执行的结果而有所不同，这样的网页被称为“动态网页”。访问动态网页的过程如下。

- (1) 客户通过浏览器向服务器申请 URL 页面。
- (2) 服务器接收请求，并处理网页上的代码。
- (3) 将代码的处理结果转换成 HTML 代码后向客户送出。
- (4) 浏览器下载并解释传来的页面并显示在浏览器中。
- (5) 服务器断开与客户的联系并转向其他客户，以便提供新的服务。

和静态网页相比，动态网页的处理上多了一个处理代码的过程。用什么方式来处理代码，在不同的历史时期采用了不同的技术，大体上可以划分为 3 个阶段。

1.1.2 动态网页发展的几个阶段

1. CGI 阶段

CGI 是英文 Common Gateway Interface 的缩写，代表服务器端的一种通用(标准)接口。每当服务器接到客户更新数据的要求以后，利用这个接口去启动外部应用程序来完成各类计算、处理或访问数据库的工作，处理完后将结果返回 Web 服务器，再返回浏览器。外部应用程序是用 C、C++、Perl、Pascal、Java 或其他语言编写的，程序运行在独立的地址空间中。具体情况如图 1.1 所示。

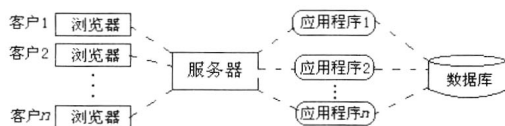


图 1.1 CGI 示意图

后来出现了 ISAPI(用于 Internet Explorer 浏览器)或 NSAPI (用于 Netscape 浏览器)技术，其功能与 CGI 相同，但技术方面有些改进。外部应用程序改用动态链接库(DLL)，被载入 Web 服务器的地址空间运行，并且用“线程”代替“进程”，因而显著地提高了运行效率。但不论是 CGI 还是 ISAPI 或 NSAPI，都需要编写外部应用程序，而编写外部应用程序并不是一件容易的事情。从开发人员的角度来讲，这种开发方式并没有带来开发上的方便。

2. 脚本语言阶段

在脚本语言阶段出现了许多杰出的脚本语言，如 ASP、PHP、JSP 等。脚本语言的出现大大简化了动态网站开发的难度，特别是 ASP 和 PHP 使用简单、功能强大，成为许多网站开发者的首选。

JSP 与 ASP 的程序结构非常相似。它的主要特点是在传统的 HTML 网页文件中加入 Java 程序段和使用各种各样的 JSP 标志(Tag)，构成 JSP 网页。Web 服务器在接收客户的访问要求时，首先执行其中的程序片段，并将执行结果以 HTML 格式返回客户。

3. 组件技术阶段

ASP.NET 和 Java(J2EE)技术是组件技术阶段的代表。这是一个由类和对象(组件)组成的完全面向对象的系统，采用编译方法和事件驱动方式运行。系统具有高效、高可靠、高

可扩展性的特点。详细情况将在下面各章节中重点讲述。

1.2 ASP.NET 发展的简要历史

进入 21 世纪前夕, 微软公司鲜明地提出了 .NET 的发展战略, 确定了创建下一代 Internet 平台的目标。什么是下一代 Internet 平台? 下一代 Internet 平台的主要特征之一就是, 它将无处不在, 世界上任何一台智能数字设备都有可能通过宽带连接到因特网上。因此, 作为下一代 Internet 平台应该实现以下要求。

- 为各种类型的客户服务。不仅能为现有的计算机、手提式计算机、移动电话等客户服务, 还要能为未来可能加入因特网的智能设备(如电视机、电冰箱、洗衣机等)提供服务。
- 强大的交互和运算能力。
- 丰富的表现能力。
- 跨平台交换数据的能力。
- 快速设计和部署的能力。
- 强有力的安全保障能力。

在这种思想的指导之下, 微软公司首先设计出了 .NET 框架(Framework), 然后在框架的基础上创建了多种应用程序的开发平台。

1.3 .NET 框架(Framework)的发展过程

1.3.1 什么是.NET 框架

什么是框架? 框架是从建筑行业引过来的名字。在建筑行业中, 它代表建筑物的骨架或基础架构。对于计算机软件来说, 框架是一簇技术的集合, 是应用程序开发的基础, 它为应用程序提供了大量的服务。如提供了运行环境和基层管理; 为应用程序的设计提供了大量可重用的类以及代码等。

Microsoft .NET Framework 框架是一个综合性的开发平台, 在这个平台之上既可以开发 Windows 桌面应用系统、也可以开发 ASP.NET Web 应用系统、Web Service 服务系统以及 Mobile 移动式(无线)网络应用系统。

当各类应用程序以 .NET Framework 框架为基础上进行开发时, 能够借助于框架提供的服务从高起点出发, 高速度地前进, 不仅可以大大简化开发过程, 还可以缩短开发周期。很多单靠应用设计本身难以解决的难题, 在 .NET Framework 服务的支持下, 都能够迎刃而解。

1.3.2 .NET 框架的发展过程

.NET Framework 的发展可以分为两个阶段。

第一阶段: 2000—2003 年微软先后推出了 .NET Framework 1.0 和 .NET Framework 1.1