



“十三五”应用型人才培养规划教材



ASP.NET 网络数据库

刘保顺 © 编著



清华大学出版社

“十三五”应用型人才培养规划教材



ASP.NET 网络数据库

刘保顺 © 编著



清华大学出版社
北京

内 容 简 介

本书介绍了在 ASP.NET 框架下采用 C# 访问 SQL Server 数据库的编程技术。主要内容包括 SQL Server 中常用的 SQL 语法,三层架构下利用 ADO.NET 的 DataAdapter、DataReader、DataSet、DataTable、Connection、Command 等对象访问 SQL Server 的过程;采用 LINQ 访问数据和数据库编程,涉及的 LINQ 技术有利用 LINQ to Object 查询内存中的集合和数据、利用 LINQ to XML 查询 XML 中的数据、利用 LINQ to Entities 操作数据库等;GridView、Chart、UpLoad、TreeView 4 种 ASP.NET 内部控件,以及 ASP.NET Web 服务从网上获取信息的技术。

除 C# 访问数据库的服务器端编程外,本书还介绍了 HTML、CSS、JavaScript、jQuery 和 jQuery EasyUI 等基础知识和客户端的编程,服务器端与客户端以 JSON 格式进行数据交换的过程,以及 jsTree 插件制作树、OrgChart 插件制作组织结构图的编程。

为避免枯燥,全书将众多网络编程的知识点融入教学示例中,并配有教学 PPT 和习题答案,以方便教学。

本书适合 ASP.NET 初学者及对 ASP.NET 感兴趣的技术人员阅读。

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 网络数据库/刘保顺编著. —北京:清华大学出版社,2019

(“十三五”应用型人才培养规划教材)

ISBN 978-7-302-52822-7

I. ①A… II. ①刘… III. ①网页制作工具—程序设计—高等学校—教材 IV. ①TP393.092.2

中国版本图书馆 CIP 数据核字(2019)第 082672 号

责任编辑:王剑乔

封面设计:刘 键

责任校对:李 梅

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

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

课件下载: <http://www.tup.com.cn>, 010-62770175-4278

印 装 者:三河市少明印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:20

字 数:483 千字

版 次:2019 年 6 月第 1 版

印 次:2019 年 6 月第 1 次印刷

定 价:59.00 元

产品编号:082226-01

网络编程涉及的知识点较多,既有后台程序的开发,也有前台界面的编程。ASP.NET 后台开发的语言目前多以 C# 为主,前台界面开发有许多优秀的框架,而 jQuery 就是最受欢迎的框架之一。目前,市面上有许多关于 ASP.NET 的书籍,其多以介绍各种控件为主,但这一编程方式由于程序深度开发时受限,如很难与 jQuery 架构整合,正在逐渐退出市场。在当今大数据时代,数据交换的格式除 XML 外,更多的是 JSON。

本书是一本介绍使用 jQuery、C# 访问和操作 SQL Server 数据库的教程。书中知识点的介绍融入教学示例中,围绕示例,循序渐进,由浅入深,尽量避免枯燥的知识性的讲解。

本书采用 Visual Studio 2012 作为程序开发平台,SQL Server 作为数据库管理系统,讲述 ASP.NET 开发 Web 程序涉及的诸多知识点,包括 HTML、CSS、JavaScript、jQuery、C#、SQL Server、ADO.NET、LINQ、Web Service 等。本书只介绍少量的内部控件如 GridView、Upload、Chart、TreeView 外,其他内部控件介绍得很少。书中以大量的篇幅介绍 ASP.NET 程序开发中浏览器端的编程知识,如 HTML、CSS、jQuery 和 jQuery EasyUI;介绍三层架构下服务器端使用 ADO.NET 对象模型、访问 SQL Server 数据库的方法;介绍浏览器端与服务器端以 JSON 格式进行数据交换的过程。

各章节安排如下。

第 1 章 ASP.NET 开发和运行环境。讲述 IIS 的安装、ASP.NET 网页程序的运行、ASP.NET 程序开发的模式、本地网页上传到云服务器等。

第 2 章 数据库基础知识。讲述关系数据库管理系统中关系模型、表间关系、建立表的原则、SQL Server 基本操作、SQL 的主要语法。

第 3 章 网页编程基础知识。讲述 HTML、CSS、JavaScript 基本语法和对象;JavaScript 解析 JSON;XML 和 AJAX 基本概念;浏览器和服务器通过 JSON 交换数据的过程。

第 4 章 ASP.NET 内置对象。介绍 Page 对象的属性和事件、Response、Request、Session、Application 和 Server 对象的属性与方法。

第 5 章 C# 语言基础知识。介绍 C# 中类的定义、封装、继承和多态性;数组和泛型的使用、服务器端 JSON 的序列化和反序列化的方法。

第 6 章 ADO.NET 连接和命令对象。介绍通过 Connection 连接 SQL Server 数据库的方法、使用 Command 和 DataReader 读取 SQL Server 表中的数据;基于三层架构,建立可重用的访问数据库的过程、注入式 SQL 攻击的防范、利用存储过程访问和操作数据库、ASP.NET 应用程序中的事务。

第7章 非连接的数据访问对象和工厂模型。介绍 DataTable 中数据的读取、删除和更新,DataSet/DataTable 的数据转换为 XML,DataSet/DataTable 对象的序列化,DataSet 中建立表间的关系,建立与提供程序无关的程序代码。

第8章 ASP.NET 内部控件。利用 Chart 将数据库中数据以图形方式展示,利用 TreeView 控件将数据库中数据以树形显示,利用 Upload 控件实现文件的上传。

第9章 jQuery。jQuery 的基本语法、jQuery 的事件、jQuery 操作 HTML 和 CSS、jQuery 操作表单、使用 jQuery 以 AJAX 方式提交表单数据、浏览器解析 XML 和 JSON 格式的数据、jquery.cookie.js、jsTree 和 OrgChart 的使用。

第10章 语言集成查询——LINQ。介绍使用 LINQ to Object 查询内存中的集合和数据、LINQ to XML 查询 XML 中数据、LINQ to Entities 操作数据库。

第11章 ASP.NET Web 服务。介绍 ASP.NET Web 服务的建立和使用、使用 Newtonsoft.Json 创建提供 JSON 格式的 ASP.NET Web 服务。

第12章 jQuery EasyUI。jQuery EasyUI 使用的方法、页面的布局、消息框和 datagrid 的使用。

本教材已列入北京科技大学校级规划教材,教材的编写和出版得到了北京科技大学教材建设经费的资助,在此深表感谢。

编著者

2019.1



本书配套教学课件、教学示例及习题答案.rar

目 录

CONTENTS

第 1 章 ASP.NET 开发和运行环境	1
1.1 Web 应用程序的工作原理	1
1.2 静态网页和动态网页	2
1.3 动态网页开发技术	4
1.4 IIS 的安装和配置	5
1.5 将本地站点上传到云	6
1.6 ASP.NET 程序开发环境和开发模式	8
1.6.1 Web 窗体拖曳控件的方式	8
1.6.2 MVC 模式	11
1.6.3 使用原生态 Web 表单模式	11
1.6.4 三层架构	16
习题和思考	16
第 2 章 数据库基础知识	17
2.1 数据库管理系统	17
2.2 数据库	18
2.3 关系数据库	18
2.4 数据库系统	19
2.5 数据库中表间的关系	20
2.6 SQL Server 的基本操作	21
2.6.1 建立数据库 students	21
2.6.2 建立数据库关系图	23
2.6.3 在数据库中增加记录	25
2.6.4 权限设置	25
2.7 关系数据库标准语言——SQL	26
2.7.1 单表数据查询	26
2.7.2 排序	28
2.7.3 统计和分组	28
2.7.4 多表数据查询	31
2.7.5 增加、删除和更新记录	32

2.7.6 操作表结构的 SQL	33
习题与思考	33
第3章 网页编程基础知识	36
3.1 HTML	36
3.1.1 HTML 文档结构	36
3.1.2 HTML 表单	38
3.1.3 表格	41
3.2 CSS	42
3.3 通过 JavaScript 为网页增加动作	46
3.4 用 JavaScript 修改 HTML 元素的样式	49
3.5 JavaScript 编写简单的扑克游戏	50
3.5.1 数组对象	51
3.5.2 自定义对象	52
3.5.3 扑克牌中的页面	54
3.5.4 扑克牌中的属性和方法	54
3.6 window 对象控制定时效果	57
3.7 XML	59
3.8 JavaScript Object Note	60
3.8.1 JSON 数据格式	60
3.8.2 JSON 文本串转换为 JavaScript 对象	63
3.8.3 将 JavaScript 对象转换为 JSON 字符串	64
3.8.4 ASP.NET 中浏览器和服务器通过 JSON 的数据交换过程	64
习题与思考	67
第4章 ASP.NET 内置对象	71
4.1 Page 对象	72
4.2 Response 对象	74
4.2.1 Response 对象的属性和方法	74
4.2.2 Response 对象应用示例	75
4.3 Request 对象	76
4.3.1 Form 集合	77
4.3.2 QueryString 集合	78
4.3.3 ServerVariables 集合	78
4.4 Server	79
4.4.1 Transfer()方法	79
4.4.2 MapPath()方法	80
4.4.3 HTML 和 URL 编码	80
4.5 ASP.NET 状态管理	80
4.5.1 ViewState 对象	80
4.5.2 Cookies	83

4.5.3	Session	86
4.5.4	Application	87
习题与思考		89
第5章 C#语言基础知识		92
5.1	类	92
5.1.1	类的定义	92
5.1.2	使用类建立对象	94
5.1.3	类的封装	97
5.1.4	类的继承	100
5.1.5	类的多态性	101
5.2	集合与泛型	103
5.2.1	泛型集合 List<T>的使用	104
5.2.2	泛型集合 Dictionary<Key, Value>的使用	105
5.3	其他数据类型	106
5.3.1	DateTime 和 TimeSpan	106
5.3.2	Convert 类	108
5.3.3	String 类	108
5.3.4	System, Text, StringBuilder 类	109
5.4	委托	109
5.5	JSON 的序列化和反序列	110
5.5.1	使用 JavaScriptSerializer 类序列化和反序列化	111
5.5.2	使用 formatter 格式化器序列化和反序列化	113
5.5.3	使用 Json, NET 序列化和反序列化	114
习题与思考		115
第6章 ADO.NET 连接和命令对象		119
6.1	ADO.NET 基础	119
6.2	Connection 对象	121
6.3	Command 类	123
6.4	DataReader 类	125
6.5	构建可重用的访问数据库的代码	127
6.5.1	数据访问层	127
6.5.2	建立连接的逻辑	130
6.5.3	在 Dbbase 类中建立查询数据的逻辑	130
6.5.4	在 Dbbase 类中建立插入数据的逻辑	132
6.5.5	在 Dbbase 类中建立更新数据的逻辑	133
6.5.6	界面层的设计	133
6.6	SQL 注入攻击	134
6.7	参数化命令	136
6.8	存储过程	137

6.8.1	建立 SQL Server 的存储过程	137
6.8.2	在 Dbbase 类中建立查询数据的逻辑	139
6.8.3	在 Dbbase 类中建立插入数据的逻辑	140
6.8.4	在 Dbbase 类中建立更新数据的逻辑	141
6.8.5	数据输入界面	141
6.8.6	显示数据页面	142
6.9	事务	143
6.9.1	存储过程事务	143
6.9.2	ADO.NET 事务	144
	习题与思考	147
第 7 章	非连接的数据访问对象和工厂模型	148
7.1	DataSet 类	149
7.2	DataSet 类的主要属性和方法	149
7.2.1	DataSet 的主要属性	149
7.2.2	DataSet 的主要方法	150
7.3	DataTable 类	152
7.3.1	DataTable 的使用	152
7.3.2	获取 DataTable 中的数据	153
7.3.3	DataTable 中删除和更新记录	155
7.4	DataAdapter 类	157
7.4.1	使用 DataAdapter 填充 DataSet	158
7.4.2	将 DataSet/DataTable 对象序列化为 XML	159
7.4.3	将 DataSet/DataTable 对象以二进制格式序列化	160
7.4.4	将 DataSet/DataTable 对象序列化为 JSON	161
7.4.5	DataSet 充填多个表和关系	164
7.4.6	利用 DataAdapter 更新数据库中的数据	166
7.5	DataView 类	167
7.6	在数据访问类中使用 DataSet	168
7.7	编写与提供程序无关的程序代码	168
7.7.1	创建工厂	169
7.7.2	使用工厂建立对象	169
7.7.3	使用与程序无关的代码查询示例	170
	习题与思考	172
第 8 章	ASP.NET 内部控件	175
8.1	GridView 控件	176
8.2	Upload 控件	180
8.3	Chart 控件	181
8.3.1	Chart 控件添加数据	183
8.3.2	Chart 控件数据绑定	183

8.3.3 制作数据回归曲线·····	188
8.4 TreeView 控件·····	194
习题与思考·····	198
第 9 章 jQuery ·····	200
9.1 jQuery 概述 ·····	200
9.1.1 jQuery 的作用 ·····	200
9.1.2 下载和引用 jQuery ·····	201
9.1.3 用 jQuery 处理 DOM ·····	201
9.1.4 显示和隐藏小狗的示例·····	201
9.2 jQuery 选择器 ·····	203
9.3 jQuery 代码执行的时机和事件 ·····	206
9.4 jQuery 动态效果的函数 ·····	207
9.4.1 显示和隐藏·····	207
9.4.2 滑动函数·····	207
9.4.3 淡入淡出函数·····	208
9.5 jQuery 对 HTML/CSS 操作 ·····	208
9.6 jQuery 操作表单 ·····	210
9.6.1 表单中元素的选择器·····	210
9.6.2 jQuery 操作表单中的元素 ·····	210
9.7 jQuery 与 Ajax ·····	213
9.7.1 Ajax 方式提交数据 ·····	214
9.7.2 浏览器解析 XML 数据 ·····	228
9.8 使用 jquery.cookie.js ·····	230
9.9 使用 jsTree 制作 tree ·····	232
9.10 使用 OrgChart 制作组织结构图 ·····	236
习题与思考·····	244
第 10 章 语言集成查询——LINQ ·····	248
10.1 LINQ to Objects ·····	249
10.1.1 LINQ 查询语法和步骤 ·····	249
10.1.2 LINQ 查询表达式 ·····	251
10.1.3 LINQ 的立即执行 ·····	254
10.1.4 方法查询 ·····	254
10.2 LINQ to XML ·····	255
10.3 LINQ to Entities ·····	258
10.3.1 生成数据模型 ·····	259
10.3.2 LINQ to Entities 查询 ·····	260
10.3.3 LINQ to Entities 数据库操作 ·····	262
习题与思考 ·····	262

第 11 章 ASP.NET Web 服务	264
11.1 Web 服务的应用	264
11.2 创建提供查询学生成绩的 Web 服务	270
习题与思考	277
第 12 章 jQuery EasyUI	278
12.1 jQuery EasyUI 概述	278
12.2 jQuery EasyUI 的 Layout	279
12.3 对话框	281
12.3.1 \$.messenger.show(options)	281
12.3.2 \$.messenger.alert(title,msg,icon,fn)	282
12.3.3 \$.messenger.confirm(title,msg,fn)	283
12.3.4 \$.messenger.prompt(title,msg,fn)	283
12.3.5 \$.messenger.progress(options or method)	284
12.4 form	284
12.4.1 form 提交数据	284
12.4.2 form 加载数据	288
12.5 jQuery EasyUI 的 Datagrid	290
习题与思考	305
参考文献	308

第1章



ASP.NET开发和运行环境



- (1) 静态网页和动态网页。
- (2) ASP.NET 开发环境。
- (3) ASP.NET 程序开发的方式。
- (4) ASP.NET 程序的运行。
- (5) 本地网页上传到云。

1.1 Web 应用程序的工作原理

1. 服务器端和客户端

Web 是基于客户端/服务器的一种体系结构。通常将提供服务的一方称为服务器,接受服务的一方称为客户端。如当用户浏览某门户网站时,该门户网站就是服务器,用户使用的计算机或者手机等智能设备就是客户端。只要在计算机上安装有提供服务的软件,这台计算机就变成一台服务器,如在计算机上装上 SQL Server 数据库管理系统,该计算机就是一台数据库服务器。Web 服务器(或 WWW 服务器)指的是装有能够接受和响应来自客户端计算机请求的特定软件的计算机,如 Apache 服务器、IIS(Internet Information Server)服务器等。在一台计算机上装上客户端软件,该计算机就成为客户端。在因特网中客户端软件一般是指浏览器软件,如微软的 IE、Mozilla Firefox、Google 的 Chrome 等。

服务器与客户端的通信过程整体可以分为以下几个过程,如图 1-1 所示。

(1) 用户在客户端浏览器输入要请求的 URL,并按 Enter 键发送这个请求。

(2) 服务器根据请求的 URL 判断客户端的请求是静态网页还是动态网页。如果请求的是静态网页,服务器找到该网页,原样送回到浏览器;如

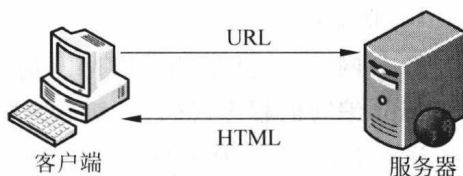


图 1-1 服务器与客户端通信过程示意图

果请求的是动态网页,服务器就会编译、执行用户请求的文件,生成标准的 HTML 文件,然后将这个 HTML 送回到客户端。

(3) 静态网页的文件由客户端上的浏览器负责解释,将解释后的结果显示在用户浏览器上。

2. 超文本传送协议(HTTP)

HTTP(HyperText Transfer Protocol)是一种以 TCP/IP 通信协议为基础的应用协议,它提供了 WWW 服务器和客户端浏览器之间传递信息的一种机制。HTTP 会话包括以下几个过程。

- (1) 客户端与服务器建立连接。
- (2) 客户端向服务器发出请求。
- (3) 如果请求被接受,则服务器将响应结果送回至客户端。
- (4) 客户端与服务器断开连接。

3. URL

因特网上每个网页都具有一个唯一的名称标识,通常称为 URL(Uniform Resource Locator,统一资源定位符)地址,这种地址可以是本地磁盘、局域网上某台计算机,也可以是因特网上某个网站,其基本格式如下。

protocol://hostname[:port]/path/

protocol: 指定的传送协议,通常为 HTTP 或 HTTPS。

hostname: 存放提供服务资源的主机名或者 IP 地址。

port: 端口号,省略时取默认值。各种传送协议都有默认的端口号,HTTP 默认端口号是 80。

path(路径): 表示主机上的一个目录或者文件名,如果省略文件名,对 IIS 服务器会查找 index.html、index.htm、index.aspx、default.html、default.htm、default.aspx(查找顺序可在 IIS 中指定)。

4. HTML

HTML(HyperText Markup Language)是超文本标记语言,标准通用标记语言下的一个应用。“超文本”就是指页面内可以包含图片、链接,甚至音乐、程序等非文字元素,其语言结构分为 head 和 body 两部分。HTML 控制文字、图片等在浏览器显示的大小、格式等,浏览器能够解释 HTML。

1.2 静态网页和动态网页

1. 静态网页

静态网页主要是由 HTML 构成。虽然有的静态网页中包含有用 JavaScript 或 VBScript 编写的程序代码(一般称为脚本),但这些脚本(Script)是随着 HTML 一起从服务器传送到客户端的,其运行是在客户端上运行的,这种脚本语言称为客户端脚本。静态网页一经制成,内容一般不会发生改变,如果要修改其内容,一般需要修改源代码,然后重新上传

到服务器。下面是一段用 HTML 编写的网页代码：

```
<html>
<head>
<title>my first web</title>
<style>
.headerH1{font-size:15pt;color:rgb(255,0,0)}
</style>
<script language="javascript">
function sayHello()
{
    alert('你好')
}
</script>
</head>
<body>
<span class="headerH1">ASP.NET 网络数据库</span>
<p onclick="sayHello()">click me</p>
</body>
</html>
```

在记事本中书写这些 HTML，保存为扩展名是 .htm 或者 .html 的静态网页。双击该文件，在 Google Chrome 浏览器中打开后效果如图 1-2 所示。这些代码包含了普通网页 HTML 的主要结构。在<head>标记中放置了<title>标记，定义网页标题，<style>标记定义了 CSS(Cascading Style Sheets, 层叠样式表)控制“ASP.NET 网络数据库”文字的大小和颜色，<script>标记标识一段 JavaScript 脚本代码。



图 1-2 用 Chrome 浏览普通网页的结果

静态网页是保存在服务器上的文件，文件间基本上是独立存在的。静态网页没有数据库支持，内容相对稳定，Web 服务器查找方便，访问效率较高。静态网页工作过程如下。

(1) 用户通过客户端的浏览器输入要访问的网页(扩展名一般是 .htm 或 .html)，发出 WWW 请求。

(2) 服务器收到静态网页的请求。

(3) 服务器从硬盘上指定的位置找到相应的 HTML 文件，发送给客户端。

(4) 客户端的浏览器收到请求的文件，并且解释这些 HTML 代码，显示出来。

需要注意的是，静态网页中并非全是 HTML，在静态网页中也可以有 JavaScript 或者

VBScript 编写的程序代码,如上面的 HTML 代码中<script>标记中的代码,当单击 click me 时会弹出一个确认框。静态网页中的代码不是在服务器上运行,而是当客户端发出 URL 请求时,服务器将代码随着 HTML 一起发送到客户端,然后在客户端运行。由于这种网页的行为是在客户端进行的,是在客户端的“动”,而网页本身是静态的,故这类网页仍然是静态网页。静态网页中的程序代码可通过 IE 浏览器中“查看”→“源”菜单命令查看。

2. 动态网页

动态网页是执行时用户可以根据输入所允许的各种信息,实现人机交互,能够根据不同的时间和地点、不同的访问者显示不同的内容,显示的内容一般来自数据库。动态网页中不仅包含 HTML 标记,还有各种相关的程序代码。动态网页工作过程如下。

(1) 用户通过客户端的浏览器输入要访问的网址,发出 WWW 请求。

(2) 服务器收到动态网页的请求。

(3) 服务器从硬盘上指定的位置找到相应的文件,并且运行这些动态网页生成标准的 HTML。

(4) 服务器将生成的 HTML 代码返回给客户端。

(5) 客户端浏览器收到这些 HTML,解释这些 HTML 代码,并显示在计算机屏幕上。

与静态网页相比,动态网页的服务器不仅要找到这些网页,而且还要在服务器上运行这些文件,生成 HTML。这样动态网页是不能在客户端通过“查看”→“源”菜单命令查看到“源代码”的,看到的是服务器运行源代码后的结果。

1.3 动态网页开发技术

动态网页是在静态网页的基础上发展起来的。第一个真正使服务器能够根据执行时的具体情况生成动态 HTML 页面的技术是 CGI(Common Gateway Interface,通用网关接口)技术,文件的扩展名为.cgi。

1994 年出现了 PHP(PHP Hypertext Preprocessor,超文本预处理程序)语言,它将 HTML 代码和 PHP 指令合成为完整的服务器端动态页面,文件扩展名为.php。

1996 年微软借鉴 PHP 思想,在其 Web 服务器 IIS 3.0 中引入了 ASP(Active Server Pages,活跃服务器页面),迅速成为 Windows 系统下 Web 服务器端的主流开发技术,文件扩展名为.asp。

1997 年 Servlet 技术问世,1998 年 JSP(Java Server Pages)技术诞生。Servlet 和 JSP 的组合让 Java 开发者同时具有了类似 CGI 程序集中处理功能和类似 PHP 的 HTML 嵌入功能,文件扩展名都为.jsp。

2000 年微软推出了基于 .NET Framework 的 ASP.NET 1.0 版本,以后新的版本不断推出,ASP.NET 可在服务器上生成功能强大的 Web 应用程序,文件扩展名为.aspx。

由于 Visual Studio 自带服务器环境,不安装 IIS 也可以在 Visual Studio 中调试 ASP.NET 程序,但如果要为客户端提供服务,如在校园网或者企业内部网上访问服务器上的网页程序,服务器上就必须安装并配置 IIS。

1.4 IIS 的安装和配置

1. 安装 IIS

ASP.NET 编写的动态网页,服务器端需要安装 IIS。Windows 默认情况下是不安装 IIS 的,如果要安装 IIS,需要 Windows 操作系统是专业版或者旗舰版。安装方法(以 Windows 7 为例):控制面板→程序→打开或关闭 Windows 功能,如图 1-3 所示。



图 1-3 IIS 的安装

在图 1-3 中,打开“Internet 信息服务”,选中“Web 管理工具”复选框,以便兼容 IIS 6。在“万维网服务”中打开“常见 HTTP 功能”,选中“静态内容”复选框。打开“应用程序开发功能”,选中 ASP.NET 复选框,如果要兼容 ASP,选中 ASP。

2. 配置 IIS

右击桌面上的“计算机”,在弹出的快捷菜单中选择“管理”→“服务和应用程序”命令,在弹出的对话框中选择“Internet 信息服务”→“连接”→“网站”命令,右击 Default Web Site,在弹出的快捷菜单中选择“添加应用程序”命令,弹出如图 1-4 所示对话框。

图 1-4 中选项说明如下。

“别名”:输入一个有意义且好记的名字,这个“别名”在运行程序时要使用,这里输入的是 example。

“物理路径”:存放网站应用程序的文件夹,可以在计算机的某盘上建立一个文件夹,将网站上的内容都放在该文件夹下。

3. 在本地计算机上运行网页程序

动态网页的程序不能双击运行。如果编程环境采用的是 Visual Studio(VS),大多数程序可以在 VS 环境中直接运行,但有些程序只能在浏览器中以 IIS 的方式运行:打开浏览

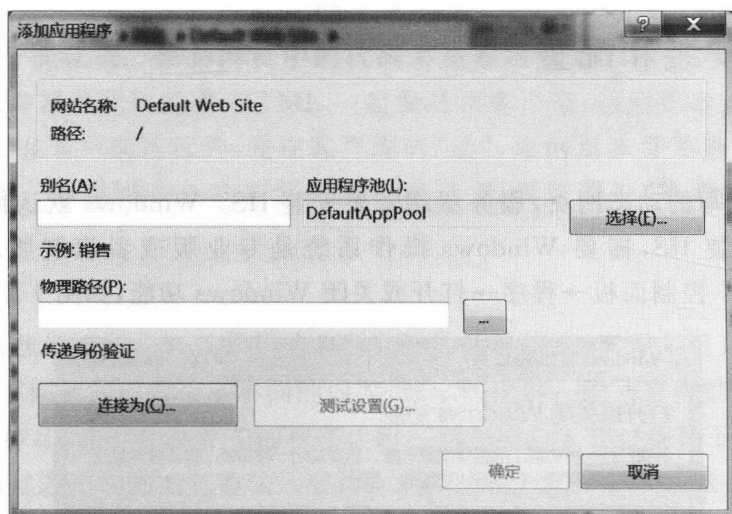


图 1-4 添加应用程序

器,在地址栏中输入“localhost:端口号/别名/网页名称”即可。

默认情况下的端口号是 80,可以省略不写,本书调试时由于 80 端口号已经分配给 PHP 程序,分配给 ASP.NET 网站的端口号是 81,如果要运行网站根文件夹下的 webform1.aspx 程序,在地址栏中输入 localhost:81/example/webform1.aspx 即可。

1.5 将本地站点上传到云

应用程序在当地计算机上调试完毕,需要发布到网上,让其他人通过 Internet 可以访问你的应用程序,最简单的方法是租用云服务。下面以租用阿里云为例,说明网站上传的过程。

登录 <https://www.aliyun.com/> 网站,网上提供的云服务器 ECS(Elastic Computer Service)是一种可伸缩性的计算服务,根据个人需要,上面有不同的产品和价格,本书租用的是 1 核 2G,1M 固定带宽,40G 高速云盘。租用 ECS 后,相当于租用了一个远程虚拟的计算机。在租 ECS 时,可以选择操作系统,对 ASP.NET 而言,需要选择 Windows Server 作为服务器。阿里云提供了公网 IP(如 47.104.91.99)和私有 IP(如 172.31.70.48)。公网 IP 用于外部访问时使用,如通过 Internet 访问数据库,连接数据库的字符串的 IP 应该使用公网 IP。如果将网页传到云上,在云上虚拟的计算机上调试网页程序,或者登录云上的 SQL Server 数据库,需要使用私有 IP。用户还需要在云上配置 ASP.NET 的环境,包括 SQL Server 数据库、IIS 等。云上有镜像市场,上面有相应的环境。在购买 ECS 后,一般可以免费得到 ASP.NET 的运行环境。当地计算机登录云的操作步骤如下。

① 在计算机“开始”→“搜索程序和文件”的文本框中输入命令: mstsc,按 Enter 键后如图 1-5 所示。

② 在图 1-5 中输入远程计算机的 IP 地址,即阿里云分配的公网 IP,单击“选项”按钮,弹出“本地资源”对话框,单击“详细设置”按钮,弹出图 1-6 所示的“本地设备和资源”的设置对话框。如果要上传的网站在当地计算机的 D 盘,就选中 D 盘,这样当进入云计算后,就在云上看到当地计算机 D 盘上的内容,可以直接将 D 盘上要发布的网站的内容方便地复制到云盘上。