

普.通.高.等.学.校
计算机教育“十二五”规划教材

ASP.NET MVC 程序设计教程

(第3版)

ASP.NET MVC PROGRAMMING
(3rd edition)

马骏 ◆ 主编



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

普通高等院校
计算机教育“十二五”规划教材

ASP.NET MVC 程序设计教程

(第3版)

ASP.NET MVC PROGRAMMING
(3rd edition)

马骏 ◆ 主编

人民邮电出版社

北京

图书在版编目 (C I P) 数据

ASP.NET MVC程序设计教程 / 马骏主编. -- 3版. --
北京: 人民邮电出版社, 2015. 8
普通高等学校计算机教育“十二五”规划教材
ISBN 978-7-115-39642-6

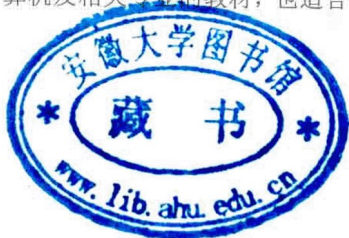
I. ①A… II. ①马… III. ①网页制作工具—程序设计—高等学校—教材 IV. ①TP393.092

中国版本图书馆CIP数据核字(2015)第163732号

内 容 提 要

本书以 VS2013 为开发环境, 介绍用 C# 和 MVC 开发 ASP.NET Web 应用程序的技术。全书分 2 篇, 第 1 篇介绍 MVC 基本编程技术, 包括 MVC 编程基础、HTML5、CSS3、JavaScript、jQuery、Bootstrap、实体框架和数据库访问技术等; 第 2 篇介绍 MVC 高级编程技术, 包括 Web API、OData、SVG、Canvas、WebGL、Three.js 等; 最后在附录中给出了与本书配套的上机练习和综合设计。

本书可作为高等院校计算机及相关专业的教材, 也适合有 C# 语言编程基础, 希望学习 ASP.NET Web 开发的人员阅读。



-
- ◆ 主 编 马 骏
责任编辑 邹文波
责任印制 沈 蓉 彭志环
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京昌平百善印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 20 2015 年 8 月第 3 版
字数: 488 千字 2015 年 8 月北京第 1 次印刷

定价: 46.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316
反盗版热线: (010)81055315

第 3 版前言

为了更好地为高校学生提供实用的、满足社会实际需求的优秀教材,为就业创造有利条件,我们在本书第 1 版《ASP.NET 网页设计和网站开发》和第 2 版《HTML5 与 ASP.NET 程序设计教程》的基础上,根据新技术的发展和社会需求,进行了大量的知识更新和改进,完成了第 3 版的编写。

本书编写思路

1. 本书第 1 版以 VS2005 为开发平台,介绍了利用 ASP.NET Web 窗体 (WebForms) 和 C#语言开发 Web 应用程序的基本设计方法;第 2 版以 VS2010 为开发平台,主要介绍了 HTML5、CSS3、JavaScript 基础以及 WebForms 的开发,同时简单介绍了利用 ASP.NET MVC3 和 Razor 视图引擎开发 Web 应用程序的基本技术;第 3 版以 VS2013 为开发平台,全面系统地介绍用 ASP.NET MVC 5 和 C#语言开发 Web 应用程序的各种编程技术,而不再介绍 WebForms 的实现。

2. 第 3 版之所以不再介绍 WebForms,是因为用 WebForms 开发 Web 应用程序项目时,主要利用的是它提供的 Web 服务器控件来实现各种功能,这在它刚推出时的 Web 应用发展环境下确实是一种创新的技术。但是,随着 HTML5 和 CSS3 正式标准的发布,直接用 HTML5 和 CSS3 实现的代码设计更简单且运行效率更高,此时 WebForms 提供的 Web 服务器控件变得越来越无用。另外,Web 服务器控件导致的性能问题、深度开发的灵活性问题、大型项目的单元测试问题都是 WebForms 无法有效解决的。而 ASP.NET MVC 是一种开源的、可扩展的 Web 开发框架,其架构的灵活性,Razor 视图引擎带来的 C#、HTML5、CSS3 和 JavaScript 混合编写动态网页的方便性,跨平台性,以及对页面和数据处理逻辑的单元测试等都是 WebForms 所无法企及的,这种新的编程模型特别适用于大型 Web 项目的开发,当然也适用于中小型 Web 项目的开发。

3. 之所以选择 VS2013 而不是选择 VS2012 或者更低版本的开发工具,是因为 VS2013 内置了 ASP.NET MVC 5 以及 HTML5 和 CSS3 正式标准的实现,而早期版本的开发工具是在 Web 正式标准推出前研制的,其内置的 HTML5 和 CSS3 验证功能和智能提示功能当然也无法完全符合新的正式标准。特别是 VS2013 提供的不同 Web 开发架构的混编技术,可以让开发人员在同一个项目中同时使用 WebForms、MVC、Web API、实时 Web 以及其他架构或者任选其中的一种或多种架构,这种涉及方方面面的大范围更新和修改是早期版本的开发工具无论如何升级都无法做到的。

4. 编写 HTML5、CSS3、jQuery、Bootstrap、JavaScript 等代码时,VS2013 提供了非常方便的智能提示功能,而早期版本的智能提示功能相对都比较弱,而且在早期版本的开发工具中属于合法但正式标准不再使用的语法也容易误导初学者,这是选择 VS2013 作为开发平台的另一个主要原因。

5. 随着天、空、地一体化需求和现代浏览器的迅速流行,以及新的 Web 正式标准的推出,早期的 Web 开发技术已经不适合当今时代的发展。作为高校学生来说,如果毕业后才发现当时所学的技术和实际需求不相符,这时就已经悔之晚矣,因此,作为本科生来说,一开始就学习成熟的新技术是提高就业能力和进一步深造的首要选择。

本书主要特点

本书是针对学习过 C#语言的读者而言的,“C#语言程序设计”是本教材的先修课程。对于没有学过 C#语言的读者,推荐先学习“十二五”普通高等教育本科国家级规划教材《C#程序设计及应用教程》(第3版,人民邮电出版社,马骏主编)一书,否则学习本书会有一些的难度。

作为 C#编程的3本系列教材之一,本书介绍的是 B/S 编程技术,如果读者希望掌握 C/S 编程技术,推荐学习工业和信息化部“十二五”规划教材《C#网络应用编程》(第3版,人民邮电出版社,马骏主编)一书。

本书主要特点如下。

1. 以目前流行的 HTML5、CSS3 和 ASP.NET MVC 为主线,选择自带 ASP.NET MVC 5 模板的 VS2013 作为开发工具,系统介绍 ASP.NET Web 应用程序开发的方法。

2. 知识点覆盖全面,教材信息量大、例子丰富、重点突出。全书基本上涵盖了 ASP.NET MVC 的各种编程技术。另外,为了让读者易理解、上手快,编者在教材结构组织、知识点的选择以及如何讲解才能循序渐进并突出重点等方面经过反复推敲、调整、增删、组合,才最终完成了本书的编写,非常适合初级 Web 编程人员学习。

3. 使用同一个项目链接本书的所有示例,让读者不但明白如何单独实现某种功能,而且还能直观地看出如何将这些技术综合到同一个 Web 应用程序项目中。

4. 通过每章的导航页分别演示不同布局页的设计和引用办法,本书基本上是一章一种新的布局。目的是为了让读者通过这些大量的布局页设计思路和具体实现,能很快举一反三,并立即将其应用到实际开发中。

5. 教材力求将晦涩难懂的技术用通俗易懂的语言表达出来,并配有大量的示例来帮助理解。读者按照本教材的顺序学习,入门快、效率高。通过阅读、理解、上机练习和调试运行,能很快掌握用 ASP.NET MVC 编写 Web 应用程序项目的各种技术。

6. 教材配套资料完整。为了配合教学需要,本书还提供了与本书配套的教学大纲、实验大纲、PPT 教学课件、习题参考解答以及所有例题、习题、上机练习和综合设计的参考源程序。

内容安排及学时分配

第1部分(第1章~第8章)是 Web 开发基础知识,如果学时有限,或者只需要学习用 C# 和 MVC 开发 Web 应用程序的基本技术,建议仅讲解这一部分。

第2部分(第9章~10章)简单介绍了 Web 开发的中、高级编程技术,这部分由于涉及其他相关知识,限于教材篇幅和授课学时限制,实际上很难在短短的两章内介绍清楚。对初学 Web 开发的学生来说,深入学习和理解这两章涉及的各种技术难度较大,只要求掌握书中介绍的基本用法即可,这是编写这两章内容的基本思路,各高校可根据先修课程的开设情况,灵活把握讲解这两章内容的深度和广度。但是,从项目应用开发的角度来说,这些中、高级内容又是必须掌握

的技术。另外，除了这两章内容外，还有一些高级开发技术本书并没有介绍，如用户访问控制与安全性处理、单元测试等。

本书所有程序均在 Windows 7 操作系统、IE 11.0 浏览器和 VS2013 简体中文旗舰版开发环境下调试通过。实际动手编写和调试程序是掌握本书知识的一个非常重要的环节，希望引起读者的高度重视。

各高校可根据实际情况，灵活调整讲授学时。各章学时分配建议如下。

54 学时				72 学时			
第 1 章	4 学时	第 9 章	2 学时	第 1 章	4 学时	第 9 章	4 学时
第 2 章	6 学时	第 10 章	4 学时	第 2 章	6 学时	第 10 章	6 学时
第 3 章	6 学时	习题课或 复习	2 学时	第 3 章	8 学时	习题课或 复习	4 学时
第 4 章	6 学时			第 4 章	8 学时		
第 5 章	6 学时			第 5 章	10 学时		
第 6 章	6 学时			第 6 章	8 学时		
第 7 章	6 学时			第 7 章	8 学时		
第 8 章	6 学时			第 8 章	8 学时		

本书由马骏担任主编，韩道军、肖春静、党兰学、杨阳担任副主编，马骏对全书进行了规划、编写、统稿、修改、增删、组合与定稿等工作。参与各章编写和配套资料整理的还有陈国强、范明虎、赵建辉、张磊、葛强、田军锋等。

由于编者水平有限，书中难免存在不妥之处，敬请读者批评指正。

编 者
2015 年 5 月

目 录

第 1 篇 ASP.NET MVC 编程基础

第 1 章 概述 2

1.1 Web 标准与 VS2013 开发环境 2

1.1.1 B/S 编程模型与 Web 标准 2

1.1.2 安装 VS2013 开发环境 4

1.1.3 ASP.NET Web 应用程序 5

1.2 创建和配置 ASP.NET MVC 5 项目 7

1.2.1 创建项目 7

1.2.2 使用 NuGet 更新程序包 10

1.2.3 修改项目配置 12

1.2.4 创建项目主页和布局页 14

1.3 本章示例的布局和创建办法 18

1.3.1 创建多个区域公用的布局页 18

1.3.2 创建本章示例使用的布局页和导

航页 22

1.3.3 添加本章示例代码 25

1.4 本书各章示例的运行说明 26

1.4.1 在桌面浏览器中观察运行结果 26

1.4.2 在手机和平板电脑模拟器中观察

运行效果 27

习题 28

第 2 章 MVC 编程预备知识 29

2.1 创建本章示例导航 29

2.2 路由及其参数传递 31

2.2.1 MVC 模式的处理过程 31

2.2.2 ASP.NET 路由 32

2.2.3 URL 模式中的参数传递 34

2.3 Razor 视图引擎 37

2.3.1 Razor 视图引擎与 ASP.NET Web

Pages 3 37

2.3.2 Razor 语法基本用法 39

2.4 用于页面全部更新的 Html 帮助器 42

2.4.1 Url 帮助器 43

2.4.2 Html 帮助器 44

2.5 用于页面局部更新的 Ajax 帮助器 44

2.5.1 Unobtrusive JavaScript Ajax 45

2.5.2 jQuery Ajax 45

2.5.3 Ajax 帮助器 46

2.6 Web 前端开发架构 (Bootstrap) 49

2.6.1 基本概念 49

2.6.2 常用的布局容器和对齐方式

CSS 类 51

2.6.3 常用的颜色组合 CSS 类 52

2.6.4 Bootstrap 栅格系统 54

2.6.5 Bootstrap 包含的图标和基本用法 57

习题 58

第 3 章 控制器、视图和模型 59

3.1 控制器和操作方法 59

3.1.1 创建本章导航 59

3.1.2 操作方法的返回类型 60

3.1.3 控制器中常用的属性和对象 65

3.2 视图 (Views) 及其分类 71

3.2.1 如何添加视图文件 71

3.2.2 布局页 (Layout Page) 73

3.2.3 视图页 (View Page) 和

视图 (View) 74

3.2.4 分部页 (Partial Page) 和分部

视图 (Partial View) 76

3.2.5 动态类型视图和强类型视图 81

3.3 模型和输入验证 83

3.3.1 定义和引用模型 83

3.3.2 绑定模型对象 85

3.3.3 利用 jQuery Validate 实现客户端验证	87	5.2.1 标题和段落	128
3.3.4 利用模型实现服务器验证	91	5.2.2 容器 (div)	130
习题	95	5.2.3 超链接	132
第 4 章 客户端脚本与事件	96	5.2.4 列表和导航 (ul、ol、dl、nav)	135
4.1 基本概念	96	5.2.5 图像、音频和视频 (img、audio、video)	137
4.1.1 文档对象模型 (DOM)	96	5.2.6 表格 (table)	140
4.1.2 JavaScript	97	5.3 表单和表单交互元素	145
4.1.3 jQuery	98	5.3.1 form 元素	145
4.2 JavaScript 代码编写基础	101	5.3.2 input 元素	151
4.2.1 数据类型和变量表示	101	5.3.3 按钮和按钮组	152
4.2.2 函数和对象	104	5.3.4 其他界面交互元素	156
4.2.3 不同类型之间的数据转换	108	5.4 表单控件帮助器及其布局方式	158
4.2.4 流程控制语句	110	5.4.1 表单控件帮助器的分类	158
4.3 利用 jQuery 操作 HTML5 元素	111	5.4.2 利用防伪标记阻止黑客攻击	159
4.3.1 jQuery 提供的基本方法	111	5.4.3 表单控件基本布局	159
4.3.2 jQuery 对象 (PlainObject) 和回调 (callback)	112	5.5 常用表单控件	162
4.3.3 元素大小和位置操作	113	5.5.1 文本框和密码框	163
4.3.4 元素的特性和属性操作	113	5.5.2 单选按钮和复选框	163
4.3.5 插入、删除、查找和替换元素	114	5.5.3 列表和下拉列表	165
4.3.6 利用 data 方法操作自定义数据	117	习题	166
4.4 浏览器窗口和客户端事件	117	第 6 章 层叠式样式表 (CSS3)	167
4.4.1 获取客户端屏幕和浏览器窗口信息	118	6.1 基本概念	167
4.4.2 客户端事件的分类	119	6.1.1 创建本章导航	167
4.4.3 使用计时器自动执行客户端代码	121	6.1.2 CSS 简介	168
习题	122	6.1.3 CSS 的级联控制	168
第 5 章 超文本标记语言 (HTML5)	123	6.1.4 CSS 的单位表示形式	171
5.1 基本概念	123	6.2 CSS 的盒模型	173
5.1.1 HTML5 简介	123	6.2.1 盒模型简介	173
5.1.2 HTML5 的基本结构	124	6.2.2 外边距、内边距和盒大小	174
5.1.3 HTML5 的全局特性	125	6.2.3 盒阴影效果 (box-shadow)	176
5.2 基本 HTML5 元素	128	6.3 CSS3 选择器	177
		6.3.1 CSS 选择器的一般格式	177
		6.3.2 基本选择器	178
		6.3.3 关系选择器	180
		6.3.4 特性选择器	184
		6.3.5 伪类选择器	185
		6.3.6 伪元素选择器	187

6.4 CSS3 样式控制	189	8.1.1 实体数据模型和实体框架 开发模式	232
6.4.1 背景图和背景渐变控制	189	8.1.2 在模型类中声明与数据库相关的 特性	233
6.4.2 显示样式控制	192	8.1.3 利用 EF6 模板和已存在的数据库 创建实体模型	234
6.4.3 字体和文本控制	193	8.2 代码优先模式完整示例	237
6.4.4 定位控制	196	8.2.1 数据库结构设计	237
6.4.5 边框控制	199	8.2.2 创建模型	238
6.4.6 伸缩盒 (flex)	200	8.2.3 添加控制器和视图	243
6.4.7 二维和三维变换控制	201	8.2.4 预处理	245
6.5 jQuery 提供的与 CSS 操作相关的 功能	204	8.2.5 课程编码管理	247
习题	207	8.2.6 基本信息管理	249
第 7 章 组件、插件和动画	208	8.2.7 成绩管理	250
7.1 基本概念	208	8.2.8 成绩查询	256
7.1.1 如何使用 Bootstrap 插件和 jQueryUI 插件	208	习题	258
7.1.2 解决 Bootstrap 和 jQueryUI 冲突的 办法	209	第 2 篇 ASP.NET MVC 高级编程	
7.1.3 Bootstrap 和 jQuery UI 提供的 选项	210	第 9 章 Web API 与 OData	260
7.2 常用组件和插件	211	9.1 基本概念	260
7.2.1 面板和嵌套面板	211	9.1.1 XML Web Service	260
7.2.2 折叠面板	212	9.1.2 Web API	261
7.2.3 对话框	214	9.2 Web API 基本设计方法	262
7.2.4 弹出框和工具提示框	215	9.2.1 JSON 对象表示法	262
7.2.5 进度条	217	9.2.2 设计和调用 Web API 服务	263
7.2.6 滑动条	218	9.3 基于 OData 的 Web API 服务	268
7.2.7 菜单和下拉菜单	218	9.3.1 什么是 OData	269
7.2.8 日期选择器	219	9.3.2 设计 Web API OData 服务	269
7.3 用 CSS3 实现复杂动画	221	9.3.3 用 jQuery ajax 调用 Web API OData 服务	270
7.3.1 CSS3 关键帧动画	221	9.3.4 用 C#调用 Web API OData 服务	272
7.3.2 CSS3 变换动画	226	习题	279
7.4 用 jQuery 实现常见动画	227	第 10 章 二维、三维图形处理 技术	280
7.4.1 jQuery 动画函数	227	10.1 SVG 和 Canvas 入门	280
7.4.2 jQuery 动画基本用法	229	10.1.1 基本概念	280
习题	231		
第 8 章 实体框架与数据库操作	232		
8.1 实体框架基础知识	232		

10.1.2 svg 元素的基本用法 283

10.1.3 canvas 元素的基本用法 287

10.2 二维图形绘制技术 290

10.2.1 矩形 290

10.2.2 圆和椭圆 291

10.2.3 直线、折线和多边形 292

10.2.4 曲线和路径 293

10.2.5 文本绘制 295

10.3 三维图形设计与实现 297

10.3.1 WebGL 和 Three.js 简介 297

10.3.2 基本用法示例 297

10.3.3 更多示例 301

附录 A 上机练习 303

A.1 上机练习要求 303

A.2 上机练习题目 304

练习 1 布局页和主页设计 304

练习 2 模型和基本操作 304

练习 3 数组排序 305

练习 4 图片浏览 305

练习 5 表单交互 306

练习 6 CSS 和动画 307

练习 7 数据库操作 308

附录 B 综合设计 309

B.1 需求说明 309

B.2 系统基本功能要求 309

B.3 源程序和文档提交要求 310

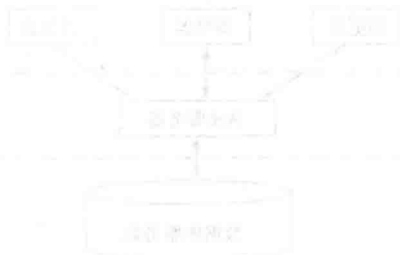
第 1 篇

ASP.NET MVC 编程基础

Web 应用程序开发分为 Web 前端开发和 Web 后端开发。Web 前端开发是指编写发送到客户端后可被各种浏览器解析的呈现给用户的页面，主要技术包括 HTML、CSS、JavaScript 等；Web 后端开发是指编写在 Web 服务器上运行的页面处理逻辑和数据处理逻辑，在 ASP.NET 开发平台上，主要用 C# 语言来实现。

ASP.NET MVC 是微软公司在 .NET 框架的基础上构建的一种免费的、开源的、基于测试驱动的 Web 应用程序编程模型。开发人员利用 ASP.NET MVC 和 Razor 视图引擎编写 Web 应用程序时，既可以混合使用 C# 代码、HTML5、CSS3 以及 JavaScript 代码快速创建前端的动态网页，又可以将后端的数据支持 (Models)、用户界面 (Views) 和逻辑处理 (Controllers) 完全分离，同时也能方便地编写单元测试代码。

这一篇我们主要学习用 ASP.NET MVC 开发 Web 应用程序的基本编程技术，这也是学习 Web 应用程序开发首先必须掌握的技术。



第 1 章 概述

作为全书的基础，这一章我们先学习 MVC 项目的创建和组织，以及各章示例的安排和运行办法。

1.1 Web 标准与 VS2013 开发环境

虽然目前存在多种典型的 Web 应用程序编程模型，但是，无论是哪种编程模型，其使用的技术都必须符合 Web 标准，这样才能在发布后被多种客户端浏览器正确识别和显示。

1.1.1 B/S 编程模型与 Web 标准

这一节我们主要学习进行 Web 开发时首先必须了解的基础知识。

1. B/S 编程模型

B/S (Browser/Server, 浏览器/服务器) 编程模型是一种以 HTTP 为基本传输协议的体系结构编程模式。在 B/S 编程模型中，开发人员只需要编写部署在 Web 服务器上的应用程序即可，而不需要编写专用的客户端程序。或者说，客户端程序是一种通过 HTTP 实现数据传输的通用应用程序，即我们平常所说的浏览器。

相对于 B/S 编程模型来说，C/S (Client/Server, 客户端/服务器) 编程模型既需要编写部署在服务器上的服务器端程序，也需要编写安装在用户计算机上的客户端程序，这种应用程序编程模型除了可以使用 HTTP 作为基础传输协议以外，还可以采用其他的网络传输协议，如 TCP、UDP 等。或者说，凡是通过下载并安装客户端程序才能运行的应用程序都属于 C/S 编程模型编写的应用程序，如 QQ、飞信、微信、360 安全卫士等都属于 C/S 应用程序。

传统 B/S 编程模型一般采用三层架构设计。用户界面、逻辑处理、数据支持构成最基本的三层架构，如图 1-1 所示。

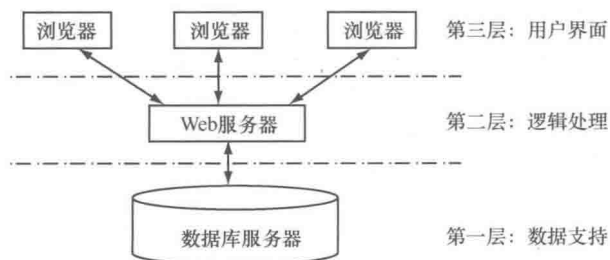


图1-1 传统B/S编程模型三层架构

2. Web 标准

Web 标准是国际上通用的 Web 设计规范,凡是符合 Web 标准规定的设计规范网站,都能用各种客户端浏览器正常访问。

(1) W3C 制定的 Web 设计标准

Web 设计标准也叫 Web 设计规范或者 Web 开发规范,这些 Web 标准大部分都是由 W3C (World Wide Web Consortium, 全球万维网联盟)和开发商以及用户等共同制定的,如 HTML、CSS、Graphics (SVG 和 Canvas API)、JavaScript、Web API、Audio and Video 以及 Mobile Web 等,W3C 都制定了对应的设计标准。

如果读者希望查看 W3C 发布的各种 Web 设计标准的具体规定和技术文档,可参考 W3C 的官方网站(英文),网址如下:

<http://www.w3.org/standards/webdesign/>

也可以参考下面的中文版网站:

<http://www.chinaw3c.org/standards.html>

但是,Web 标准只提供了功能和设计规范,而具体实现则由软件生产厂家来完成。换言之,标准仅说明了可以使用哪些功能,这些功能的语法格式,以及在使用符合标准的内容时,哪些是推荐使用的,哪些是建议不要这样用的。而如何实现符合标准规定的内容,则由具体的开发工具来决定。

(2) Web 1.0 和 Web 2.0

从大的方面来看,Web 开发经历了从 Web 1.0 到 Web 2.0 的变迁。随着以 HTML5 为核心的 Web 2.0 时代的到来,以及“天、空、地”一体化的各种 Web 应用,Web 设计标准和相应的实现技术也都发生了翻天覆地的变化。

1999 年,W3C 制定了 HTML 4.01 标准,随后公布了 CSS 2.1 标准和 JavaScript 标准。这时的 HTML、CSS 以及 JavaScript 标准统称为 Web 1.0 标准。

HTML5、CSS3、JavaScript、Canvas、SVG、WebGL 以及移动设备开发规范等都是 W3C 发布的新一代 Web 开发标准,为了将其和早期的设计标准区分开,一般将这些新的设计标准统称为 Web 2.0 标准。

随着 HTML5、CSS3 等新标准的正式发布和快速流行,不支持这些标准的旧版浏览器正在迅速被淘汰,为了在客户端浏览器的残酷竞争中占有一定的用户量,各大浏览器生产厂家都在快速更新所提供的浏览器版本,以尽快实现新的 Web 标准,避免被其他浏览器淘汰。

目前世界上流行的各种“现代浏览器”的最新版本都支持新的 Web 标准,如微软公司的 IE 11.0 浏览器、Mozilla 基金会的 FireFox 浏览器、谷歌公司的 Chrome 浏览器以及 Opera 浏览器、Safari 浏览器等。

在 VS2013 下编辑 HTML 文档或者 CSS 时,系统会自动检查所编辑的内容是否符合 HTML5 正式标准和 CSS3 正式标准(正式标准不再包含各厂家自定义的前缀)。当开发人员编写的代码不符合正式标准规定的规范时,编辑器会自动显示绿色的波浪形下划线提醒 Web 开发人员。

3. 静态网页和动态网页

许多初学者都误将包含各种动画、滚动字幕等视觉上具有“动态效果”的网页认为是动态网页,否则认为是静态网页,其实这样理解是不正确的。实际上,无论是动态网页还是静态网页,都可以展示文字、图片、动画等动态效果,但从网页生成的内部方式来看,静态网

页和动态网页却有着本质的差别。

静态网页是指 Web 服务器发送到客户端的静态 HTML 页面，其特点是 URL 固定、内容相对稳定、容易被搜索引擎检索。在静态网页上，一样可以出现各种动态的效果，如动画、滚动字幕等。图 1-2 展示了静态网页的基本工作原理。

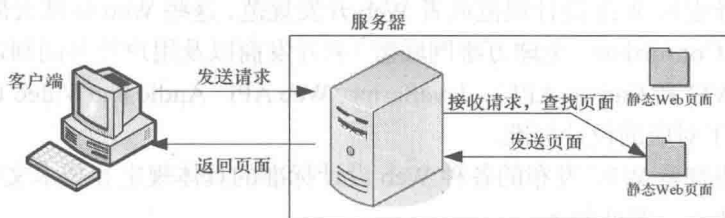


图1-2 静态网页工作原理

动态网页是指 Web 服务器根据客户端请求，随不同用户、不同时间的操作，动态返回不同静态内容的网页。换言之，当客户端向服务器发送请求时，服务器先对其进行处理，然后再将处理结果转换为静态网页发送到客户端。图 1-3 展示了动态网页的基本工作原理。

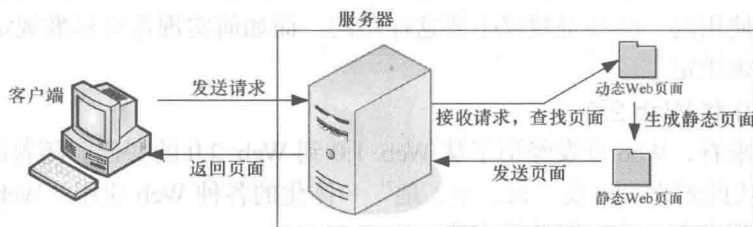


图1-3 动态网页工作原理

采用动态网页技术的网站可以实现比静态网页更多的功能。

1.1.2 安装 VS2013 开发环境

Microsoft Visual Studio 系列开发工具是 Windows 操作系统平台上最流行的开发工具，它提供的各种方便的开发模型使开发人员能快速地构建和编写各种复杂的应用程序，或者说，只需要使用同一种开发工具，就能完成所有的软件开发任务。

本书介绍的所有内容都是在 VS2013 开发环境下用【ASP.NET Web 应用程序】模板来实现的。所有源程序都在 Windows 7 操作系统和 IE 11.0 浏览器下调试通过。

1. 安装包含 Update4 的 VS2013 旗舰版

VS2013 内置了 HTML5、CSS3 以及 ASP.NET 5.0 等相关技术，特别是它的“模型联编”技术、HTML5 和 CSS3 等正式标准设计规范的检查，以及非常方便的智能提示等，十分便于 Web 应用程序的开发。这些集成技术带来的便捷是 VS2012 以及更早版本所不具备的。学习本书内容前，应先安装 VS2013 开发工具。

本书开发环境的具体要求如下。

(1) 操作系统：Windows 7 (32 位或者 64 位)。

(2) 内存：最低 1GB，建议 2GB 或者更高。

(3) 包含 Update4 的 VS2013 旗舰版 (Visual Studio Ultimate 2013 with Update 4)。

由于 VS2013 的安装过程比较简单，这里不再介绍具体安装步骤。

如果不使用 VS2013 旗舰版，也可以安装包含 Update4 的 VS2013 开发版，这是一种免费的版本，主要用于个人学习和小公司使用，但其功能没有 VS2013 旗舰版强大。

2. 下载和安装 IE 11.0 浏览器

IE 11.0 浏览器全面实现了 HTML5、CSS3、JavaScript、Canvas、SVG 以及 WebGL 等 Web 2.0 正式标准的功能，而且运行速度快、使用方便，而 IE 9.0 和 IE 10.0 只是实现了 Web 2.0 推荐标准草案中的部分功能，IE 8.0 和 IE6.0 由于研发时间更早，不支持 Web 2.0 标准。

IE 11.0 浏览器的官方下载网址如下：

<http://windows.microsoft.com/zh-cn/internet-explorer/ie-11-worldwide-languages>

从该下载网页选择合适的 IE 11.0 版本（32 位或者 64 位简体中文版，网站会根据访问时使用的操作系统自动选择合适的版本），下载后直接运行安装即可。

至此，我们完成了本书需要的开发和调试环境的安装。

1.1.3 ASP.NET Web 应用程序

ASP.NET 是一个免费的 Web 开发平台，是微软公司在 .NET 框架的基础上构建的一种 Web 开发架构。2014 年 11 月 12 日，微软公司在纽约举办的全球开发者联盟大会上郑重声明，不但现在的 ASP.NET 是完全免费的，以后所有 .NET 框架也将全部开源（可支持 Windows、Mac OS X 和 Linux 三大操作系统）。关于微软 .NET 框架开源的更多信息，有兴趣的读者可参考下面的网站（或者搜索“.NET 开源”查看国内翻译版）：

<http://winsupersite.com/visual-studio/visual-studio-and-net-go-cross-platform>

1. One ASP.NET

VS2013 提供的 ASP.NET Web 应用程序模板同时提供了多种 Web 应用编程模型，从大的方面来说，这些编程模型主要分为两大类，一类是网站（Sites）编程模型，另一类是服务（Services）编程模型，如图 1-4 所示。

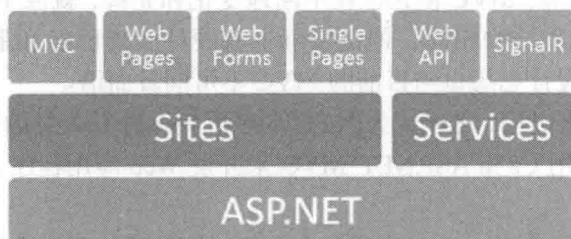


图1-4 ASP.NET Web应用程序编程模型

这种在同一个 ASP.NET Web 应用程序项目中可同时选择多种 Web 编程模型的技术称为“模型联编”技术，也叫“One ASP.NET”，这也是从 VS2013 开始首次使用的技术。

“One ASP.NET”的本质含义为：在【ASP.NET Web 应用程序】模板提供的编程模型中，既可以仅选择一种模型，也可以根据项目需要同时选择多个模型，包括 MVC（扩展名为.cshtml 的文件）、Web Pages（扩展名为.html 的文件）、Web Forms（扩展名为.aspx 的文件）、Single Pages（移动设备常用的单页应用程序开发模型）、Web API（基于 HTTP 的 Web 服务编程接口）以及 SignalR（支持云环境的实时 Web 应用服务，如天气预报服务、地图服务等）。

在这些编程模型中，最常用的有两种，一种是基于事件驱动的编程模型（简称 WebForms 或 Web 窗体），另一种是基于测试驱动的编程模型（简称 MVC）。

2. Web 窗体编程模型

在 VS2013 开发工具推出之前, .NET 开发人员主要用【ASP.NET Web 窗体应用程序】模板来开发 Web 应用程序, 这是一种基于事件的敏捷 Web 开发模型。从 VS2013 开始, 微软公司不再专门提供【ASP.NET Web 窗体应用程序】模板, 但仍然在开发工具中保留了【Web 窗体】开发模式。

【Web 窗体】的最大优势是易理解、上手快, 非常适合利用它内置的 Web 服务器控件开发 Web 应用程序。但是, 随着 Web 2.0 时代的到来, 新的 Web 标准使 HTML5 和 CSS3 的功能变得非常强大, 此时【Web 窗体】的缺点也逐渐暴露出来(这也恰恰是它刚推出时的优点), 最突出的问题主要体现在以下两个方面。

(1) 新版本的 Web 窗体为了兼容旧版本, 使 Web 服务器控件封装的功能变得越来越臃肿, 这在一定程度上影响了程序运行的效率。另外, HTML5、CSS3 以及各种优秀开源架构的推出, 也让【Web 窗体】提供的 Web 服务器控件变得越来越无用。

(2) 随着大型 Web 应用程序项目的分工越来越细, 【Web 窗体】编程模型使大型 Web 项目的单元测试工作变得非常棘手。

在这种情况下, 微软公司又推出了开源的、基于测试驱动的【MVC】编程模型。

3. MVC 编程模型

ASP.NET MVC 是微软公司实现的、开源的、基于测试驱动的 Web 应用程序编程模型。官方网址如下:

<http://www.asp.net/mvc>

MVC (Models、Views、Controllers) 其实并不是一种新技术, 该设计思想早在 20 世纪 70 年代就已经被提出。MVC 体系结构的精髓是它可以帮助开发人员创建数据支持 (Models)、逻辑处理 (Controllers) 和用户界面 (Views) 完全分离的应用程序, 由于这种物理上的结构分离非常清晰, 所以即使系统非常庞大, 也同样容易维护和扩展。

随着 Web 技术的发展, MVC 模式也在不断地变化和完善, 就像 1946 年的第 1 台计算机与目前的计算机虽然基本结构相同, 但是实际上其硬件实现已经和最初的结构千差万别类似, 最初提出的 MVC 设计思想也与目前的 MVC 不可同日而语。

ASP.NET MVC 推出以后, 迅速成为 .NET 环境下 Web 开发人员的首选技术。目前大型软件公司都建议开发人员用 C# 和 ASP.NET MVC 来开发 Web 应用程序, 而不再建议用传统的 Web 窗体技术来开发。

不过, 对于微软公司来说, 为了让开发人员仍然能继续使用原来已经熟悉的技术, 在其提供的开发工具中, 【Web 窗体】和【MVC】这两种编程模型将会一直同时存在。

(1) MVC 编程模型的特点

ASP.NET MVC 编程模型具有以下特点。

① 任务分离。

ASP.NET MVC 强制实施“任务分离”的策略。在任务分离的过程中, 应用程序被分成离散的松耦合部件, 即: 模型、视图和控制器。这种任务分离模式使复杂 Web 应用程序更易于测试、维护和扩展。

在 VS2013 开发环境下, ASP.NET MVC 中的模型 (Models) 用于存放独立且可重复使用的组件, 用 C# 实现的组件默认都保存在项目的 Models 文件夹下; 视图 (Views) 用于呈现网页界面, 用 Razor 视图引擎、HTML5、CSS3 以及 JavaScript 等实现的视图页面默认都保存在

项目的 Views 文件夹下；控制器（Controllers）用于控制整个网站的处理流程以及协调 Views 和 Models 之间的数据传递，用 C#实现的这些代码默认都保存在项目的 Controllers 文件夹下。

② 基于测试驱动的开发。

利用 ASP.NET MVC 架构，可以对控制器中提供的每个操作方法都单独进行单元测试，从而为测试驱动的开发（TDD）提供了更好的支持，非常适合大型团队开发功能和层次结构都很复杂的 Web 应用程序。

③ 对 HTML5 和 CSS3 的操控能力高。

VS2013 内置的 ASP.NET MVC 5 显著提高了开发人员对 HTML5 和 CSS3 的操控能力，该架构完全公开和实现了 HTML5 和 CSS3 规范，这是【Web 窗体】模型无法做到的。换言之，早期的【Web 窗体】模型在简化功能设计的同时，也导致开发人员降低了对 HTML5 和 CSS3 进行深度控制或者细粒度控制的灵活性。

（2）ASP.NET MVC 的版本发展

本书成稿时，ASP.NET MVC 的稳定版是第 5 版，该版本也是 VS2013 内置的版本。

ASP.NET MVC 的发展历程如下。

2008 年，微软公司推出了 .NET 框架 3.5 和 VS2008。VS2008 首次内置了 ASP.NET MVC 1.0。但是，该版本仅支持 ASPX 视图引擎，而且功能有限。

2010 年，微软公司推出了 .NET 框架 4.0 和 VS2010。VS2010 内置了 ASP.NET MVC 2，该版本仍然是仅支持 ASPX 视图引擎。

2012 年，微软公司推出了 .NET 框架 4.5 和 VS2012。在 VS2012 开发环境中，同时内置了 ASP.NET MVC 3（简称 MVC3）、ASP.NET MVC 4（简称 MVC4）以及 ASP.NET Web API 等架构。从 MVC3 开始，首次引入 Razor 视图引擎，而且其后续的更高版本默认也都使用 Razor 视图引擎。

2013 年，微软公司推出了 .NET 框架 4.5.1 和 VS2013，并在 VS2013 开发环境中内置了 ASP.NET MVC 5（简称 MVC5），同时将 Web 窗体、MVC、Web API、实时 Web 以及实体框架等架构全部整合在一起。

总之，VS2013 开发环境突出了 ASP.NET 模型联编的概念，在 VS2013 的 Web 选项卡中，只有一个【ASP.NET Web 应用程序】模板，在该模板中，开发人员可根据业务需要，在同一个解决方案或者同一个项目中任选【MVC】、【Web 窗体】、【Web API】等编程模型之一，也可以同时选择这些编程模型。

1.2 创建和配置 ASP.NET MVC 5 项目

安装 VS2013 开发环境后，可直接创建【ASP.NET Web 应用程序】项目。为了方便读者学习，并且对项目的模块组织和完整实现有一个全面的了解和把握，本书配套的所有源程序都将包括在这一节创建的 Mvc5Examples 项目中。

1.2.1 创建项目

这一节我们学习如何创建新项目，并配置项目中使用的程序包版本。