

计算机科学与技术规划教材 信息系统方向

# Web应用开发技术

石双元 编著



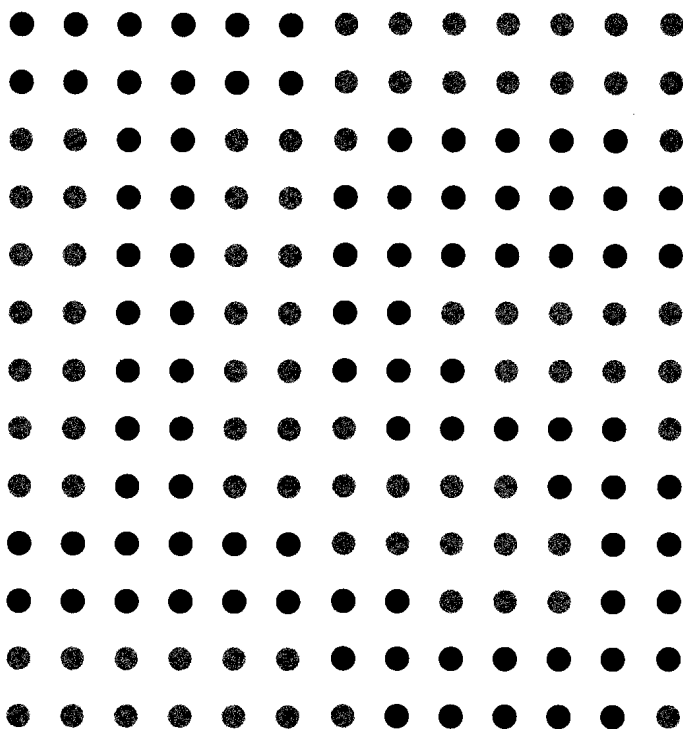
清华大学出版社

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

计算机科学与技术规划教材 信息系统方向

# Web应用开发技术

石双元 编著



清华大学出版社  
北京

## 内 容 简 介

本书以 Microsoft ASP.NET 和 C# 语言为背景,力图系统、全面地介绍 Web 应用开发所涉及的内容和最新进展。本书围绕 Web 应用开发所涉及的各项技术由浅入深地展开,在内容和结构安排上力求做到系统性和连贯性。

本书共 15 章,分为 3 部分。第一部分包括第 1~4 章,主要介绍 Web 应用与开发的基本概念与特征、C# 语言的基础知识以及面向对象特性在 C# 语言中的表现形式和实现方法、Web 开发的背景知识等。

第二部分包括第 5~11 章,全面介绍了 Microsoft ASP.NET 的框架与控件和基于 ADO.NET 的数据库开发技术。分别阐述了 ASP.NET 页面程序结构、Web 服务器控件、ASP.NET 中的数据绑定技术以及控件与代码分离技术。

第三部分包括第 12~15 章,作为高级应用部分,重点介绍了作为客户端开发的主流技术 JavaScript 及其对象,融入最新 Ajax 技术的原理和常用的框架与控件,如 Microsoft ASP.NET Ajax。

本书可作为高等院校信息管理与信息系统、电子商务、工商管理及管理理学等专业的教材,也可供相关专业科技人员、工程技术人员和其他人员参考。

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

Web 应用开发技术 / 石双元编著. —北京:清华大学出版社,2010.9

(计算机科学与技术规划教材 信息系统方向)

ISBN 978-7-302-22655-0

I. ①W… II. ①石… III. ①主页制作—程序设计 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2010)第 083703 号

责任编辑:张瑞庆 顾 冰

责任校对:李建庄

责任印制:王秀菊

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,jsjic@tup.tsinghua.edu.cn

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

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185×260

印 张:28.75

字 数:716 千字

版 次:2010 年 9 月第 1 版

印 次:2010 年 9 月第 1 次印刷

印 数:1~3000

定 价:39.00 元

---

产品编号:034106-01

# 前 言

Web 应用开发是近年来随着 Web 应用的普及和深入而形成的研究 Web 信息系统开发理论与方法的综合性技术。目前应用系统有两种主要模式,一种是基于 C/S 结构的桌面应用,另一种就是基于 B/S 结构的 Web 应用。近年来,Web 应用模式所占的比重在逐年增长,尽管在近期内不可能完全取代桌面应用,但它必将是未来企业应用系统的主流和发展方向。

根据不同的应用领域,Web 应用可分为公众型 Web 应用和企业型 Web 应用。Web 技术在以 Google、Flickr、Amazon 和 Facebook 等为代表的公众型应用领域中取得了巨大成功,并已开始向以桌面应用为主的企业应用领域渗透,企业应用平台从桌面应用向 Web 迁移的趋势不可逆转,传统的桌面应用正面临 Web 的巨大挑战。因此,Web 应用开发技术是衡量信息管理方向学生能力的一个重要指标,在很多大公司的招聘考试中,Web 技术占很大的比例。

根据不同的开发与应用平台,Web 应用开发又有两大分支,一是基于 Java 平台,二是基于 Microsoft ASP.NET 平台。不过,这两种平台也有很多的共性,都涉及 HTML、XML 和数据库技术等,并且其客户端技术正在融合。

Web 应用开发技术是一门综合性技术课程,它涉及的内容非常多,如 HTML、CSS、C#(或 Java、PHP 等)、JavaScript、XML、JSON、DOM 及其最新的 Ajax 框架与控件等,并且这些技术是相互关联和依存的,其中的每一项技术涉及的内容也非常丰富。目前,介绍 Web 开发技术方面的书籍很多,但其中绝大多数是突出某一个单项技术,这些书籍作为教材不太合适,因为无法使学生系统、全面地了解 Web 应用开发技术的全貌。这也是编写本书的难点所在,Web 应用涉及的技术太多,要在有限的篇幅和课时限制的条件下全面安排这些内容是一件非常困难的事情,尽管我们力求使其合理,但不得不舍弃很多。

本书以 Microsoft ASP.NET 框架和 C# 语言为背景,选择 C# 作为服务器端语言,不仅因为 C# 语言是为在 .NET 中使用和支持基于组件的编程而特别设计的,更因为其完全面向对象和简单易学。本书力图系统、全面地介绍 Web 应用开发所涉及的内容和最新进展,围绕 Web 应用开发所涉及的技术由浅入深地展开。在内容和结构安排上力求做到系统性和连贯性。首先讲述了面向对象技术的基本概念和特性;接着简要介绍了 C# 语言的基础知识以及面向对象特性在 C# 语言中的表现形式和实现方法,为了避免重复,将 C# 的类的设计和高级编程技术等内容融合到后面的代码分离技术和自定义控件的设计中;然后对 ASP.NET 开发涉及的背景知识进行了简单的介绍,以使没有相关背景知识的读者也能很快熟悉和掌握这些知识(如果课时紧张,背景知识部分可作为选学内容或自学内容)。在此基础上完整介绍了 Microsoft ASP.NET 的框架、服务器控件和基于 ADO.NET 的数据库

开发技术。作为高级应用部分,本书还介绍了作为客户端开发的主流技术 JavaScript 及其对象,融入最新 Ajax 技术的原理和常用的框架与控件,如 Microsoft ASP.NET Ajax。

全书分为 15 章。第 1 章主要介绍了 Web 应用开发的现状、前沿及其特征;第 2、3 章讨论了 C# 语言的基础知识和面向对象方法在 C# 语言中的实现;第 4、5 章讲述了 ASP.NET 开发所需的背景知识和基本方法,包括 HTML、XML 和 ASP 等;第 6~11 章以 C# 语言为支持语言,用大量实例系统地介绍了 ASP.NET 的开发方法与技术,其中,第 6、7 章介绍 ASP.NET 的基础控件、验证控件与扩展控件,第 8、9 章介绍基于 ADO.NET 的数据连接、数据绑定技术及各种列表控件与数据绑定的实现方法,第 10 章讨论了 ASP.NET 中的代码分离技术,第 11 章介绍了如何在 ASP.NET 中应用 XML 技术;第 12、13 章介绍了 JavaScript 基础知识及对象;第 14、15 章介绍了 Ajax 技术及 Microsoft ASP.NET 框架与控件。为便于初学者掌握,本书每章都有大量的实例程序。

通过本书不仅可以系统地学习目前最新的 Web 应用开发技术与方法,而且可以了解和掌握 Web 开发的最新技术,如页面与代码分离、JavaScript 对象、JSON 描述方法和 Ajax 技术。

本书可作为高等院校信息管理与信息系统、电子商务、工商管理及管理学等专业的教科书和参考书,还可供有关领域的科学工作者以及系统开发人员阅读参考。

本书由华中科技大学石双元和刘伟编写。石双元负责全书的策划、统稿以及全书格式和文字核对工作。研究生张巨松在文字核对方面做出了大量的工作,在此表示感谢。本书的编写得到了华中科技大学管理学院蔡淑琴教授的大量帮助和指点,以及清华大学出版社有关同志的大力支持和帮助,在此对他们表示诚挚的谢意。在本书的编写过程中,我们参考和引用了不少国内外文献资料,在此对这些资料的作者表示深深的感谢。

由于本书的写作时间较短,以及编者水平有限,书中难免存在缺点和错误,敬请各位专家和读者批评指正。

石双元

# 目 录

<b>第 1 章 绪论</b> .....	1
1.1 Web 应用及其特征 .....	1
1.1.1 什么是 Web 应用 .....	1
1.1.2 Web 网站与 Web 应用 .....	2
1.2 Web 应用的相关技术 .....	3
1.3 Web 应用的分类及其差异 .....	5
1.3.1 用户群体与使用行为的差异 .....	5
1.3.2 用户使用模式的差异 .....	5
1.3.3 业务处理模式的差异 .....	6
1.3.4 企业级 Web 应用与桌面应用的差异 .....	6
1.4 企业级 Web 应用及关键问题 .....	7
1.4.1 企业应用从桌面向 Web 迁移的必然性 .....	7
1.4.2 企业环境中的 Web 应用的成功示例 .....	8
1.4.3 企业级 Web 应用开发所面临的挑战 .....	8
<b>第 2 章 C# 语言基础</b> .....	10
2.1 C# 语言及其特点 .....	10
2.2 中间语言代码与 .NET 的跨平台 .....	11
2.3 一个简单的 C# 程序 .....	12
2.4 命名空间 .....	16
2.4.1 什么是命名空间 .....	16
2.4.2 命名空间的引用 .....	17
2.5 C# 程序的构成 .....	18
2.6 C# 的类和对象定义 .....	19
2.7 C# 的格式与数据类型 .....	22
2.7.1 标识符 .....	22
2.7.2 书写规则 .....	23
2.7.3 C# 的数据类型 .....	24
2.7.4 数据类型转换 .....	25

2.7.5	常量	27
2.7.6	变量	28
2.7.7	枚举与数组	29
2.8	运算符与表达式	32
2.8.1	运算符	32
2.8.2	表达式	35
2.9	方法定义与使用	36
2.10	定义方法	38
2.11	流程控制语句	39
2.11.1	结构化程序设计的3种基本流程	39
2.11.2	分支语句	39
2.11.3	循环语句	42
2.11.4	跳转语句	44
2.11.5	递归调用	46
<b>第3章</b>	<b>C#的面向对象特性</b>	<b>48</b>
3.1	类的定义与C#修饰符	48
3.1.1	类的定义	48
3.1.2	C#修饰符	50
3.2	创建对象与构造方法	51
3.2.1	创建对象	51
3.2.2	构造方法	52
3.2.3	析构方法	55
3.2.4	类的修饰符	56
3.3	类的成员及其修饰符	60
3.3.1	变量及其修饰符	60
3.3.2	方法及其修饰符	65
3.4	访问控制符	71
3.4.1	公共访问控制符 public	71
3.4.2	默认访问控制符	72
3.4.3	保护访问控制符 protected	72
3.4.4	私有访问控制符 private	72
3.5	C#的继承及其实现	73
3.5.1	继承的定义	73
3.5.2	继承的实现	75
3.6	C#语言中的多态与实现	77
3.6.1	什么是多态	77
3.6.2	构造方法的继承与重载	79
3.6.3	C#多态的实现	80

3.7	接口及其实现	82
3.7.1	单重继承与多重继承	82
3.7.2	C# 接口概述	82
3.7.3	接口的定义	83
3.7.4	实现接口	84
3.8	异常处理	85
3.8.1	异常产生的原因	85
3.8.2	捕获异常与抛弃异常	86
<b>第4章</b>	<b>Web 应用基础知识</b>	<b>88</b>
4.1	HTML 概要	88
4.1.1	HTML 文档的基本结构	88
4.1.2	HTML 文档结构标签	90
4.1.3	版面风格控制	92
4.1.4	HTML 中显示特殊字符	97
4.1.5	文件的链接	99
4.1.6	HTML 表单	101
4.1.7	HTML 样式表	105
4.2	XML 结构及语法	111
4.2.1	XML 及其特征	111
4.2.2	XML 文档结构	113
4.2.3	XML 文档格式定义	116
4.2.4	XML 文档的显示与转换	119
4.3	ASP 动态页面技术	124
4.3.1	静态网页与动态网页	124
4.3.2	ASP 及其特征	125
4.3.3	ASP 文件结构	126
4.3.4	ASP 页面的运行环境	127
4.3.5	VBScript 语法简介	128
4.3.6	ASP 与 ADO 的连接	133
4.3.7	ASP 内置对象	138
4.3.8	JavaScript、JScript、VBScript 的简要说明和比较	143
<b>第5章</b>	<b>ASP.NET 概览</b>	<b>145</b>
5.1	ASP.NET 及其特征	145
5.1.1	从 ASP 到 ASP.NET	145
5.1.2	ASP.NET 及其主要特性	146
5.1.3	.NET 框架概述	147
5.2	ASP.NET 环境配置	148



5.2.1	Web 服务器安装与配置 .....	149
5.2.2	ASP.NET 开发环境的配置 .....	154
5.2.3	ASP.NET 常用文件类型 .....	154
5.3	ASP.NET 中的类与命名空间 .....	155
5.3.1	ASP.NET 中的类 .....	155
5.3.2	ASP.NET 的命名空间 .....	156
5.4	ASP.NET 页面表单与服务器控件 .....	157
5.4.1	ASP.NET 页面表单 .....	157
5.4.2	服务器控件 .....	157
5.5	ASP.NET 页面 .....	159
5.5.1	简单的 ASP.NET 页面 .....	159
5.5.2	ASP.NET 页面的程序结构 .....	161
5.6	控制页面导航 .....	167
5.6.1	参数的传递与接收 .....	167
5.6.2	页面之间的数据共享 .....	168
5.7	Web 应用中的 WebService .....	170
<b>第 6 章</b>	<b>ASP.NET 常用控件 .....</b>	<b>174</b>
6.1	Web 基础控件介绍 .....	174
6.1.1	Label 控件 .....	175
6.1.2	Button 控件 .....	176
6.1.3	TextBox 控件 .....	178
6.1.4	DropDownList 控件 .....	180
6.1.5	ListBox 控件 .....	184
6.1.6	CheckBox 控件 .....	187
6.1.7	CheckBoxList 控件 .....	188
6.1.8	RadioButton 控件 .....	190
6.1.9	RadioButtonList 控件 .....	191
6.1.10	HyperLink 控件 .....	194
6.1.11	列表类控件的属性与方法 .....	195
6.2	ASP.NET 基础控件综合应用 .....	198
6.3	ASP.NET 验证控件 .....	199
6.3.1	表单信息的验证 .....	199
6.3.2	RequireFieldValidator 控件 .....	200
6.3.3	RegularExpressionValidator 控件 .....	202
6.3.4	CompareValidator 控件 .....	205
6.3.5	RangeValidator 控件 .....	208
6.3.6	ValidationSummary 控件 .....	209
6.3.7	CustomValidator 控件 .....	212

<b>第 7 章 ASP.NET 高级控件与扩展控件</b> .....	<b>215</b>
7.1 ASP.NET 高级控件 .....	215
7.1.1 Panel 控件 .....	215
7.1.2 通过编程控制页面上的控件 .....	221
7.1.3 Calendar 控件 .....	223
7.1.4 Input(File)控件 .....	224
7.2 ASP.NET 扩展控件 .....	224
7.2.1 IEWebControls 的安装 .....	225
7.2.2 树形视图 TreeView 控件 .....	226
7.2.3 工具条 ToolBar 控件 .....	232
7.2.4 选项 TabStrip 控件 .....	234
<b>第 8 章 ADO.NET 与数据绑定</b> .....	<b>238</b>
8.1 什么是 ADO.NET .....	238
8.2 ADO.NET 的访问方法 .....	239
8.3 ASP.NET 中的数据库操作 .....	242
8.3.1 检索数据库中的记录 .....	242
8.3.2 检索单条记录 .....	243
8.3.3 在 ASP.NET 中实现动态查询 .....	244
8.3.4 向数据库中添加记录 .....	246
8.4 在查询中使用参数 .....	247
8.5 使用 SQL 存储过程 .....	249
8.6 ASP.NET 中的数据绑定 .....	252
8.6.1 什么是数据绑定 .....	252
8.6.2 绑定数据源到列表控件 .....	252
8.6.3 利用数据源控件绑定数据 .....	257
<b>第 9 章 ASP.NET 数据控件</b> .....	<b>259</b>
9.1 ASP.NET 的数据控件介绍 .....	259
9.2 用 Repeater 控件显示数据 .....	264
9.2.1 绑定 Repeater 控件到数据源 .....	264
9.2.2 在 Repeater 控件中使用模板 .....	266
9.2.3 在 Repeater 控件内绑定其他控件 .....	267
9.2.4 用 Repeater 控件实现主从表 .....	269
9.3 用 DataList 控件显示数据 .....	271
9.4 用 GridView 显示数据 .....	275
9.4.1 绑定 GridView 控件到数据源 .....	277
9.4.2 在 GridView 控件中创建列 .....	279

9.4.3	在 GridView 控件中实现排序 .....	285
9.4.4	DataSet 的使用技术 .....	287
9.4.5	在 GridView 中实现分页 .....	289
9.5	Repeater、DataList 与 GridView 比较 .....	292
<b>第 10 章</b>	<b>程序代码与页面分离技术 .....</b>	<b>294</b>
10.1	创建和使用 .NET 组件 .....	294
10.1.1	业务组件及其优势 .....	294
10.1.2	ASP.NET 组件的建立 .....	295
10.1.3	ASP.NET 组件的使用 .....	297
10.1.4	使用和访问组件的属性 .....	298
10.2	代码隐藏技术 .....	302
10.3	用户控件 .....	305
10.3.1	用户控件的定义与使用 .....	305
10.3.2	使用用户控件的属性与方法 .....	308
10.4	自定义控件 .....	310
10.4.1	自定义控件及其特性 .....	310
10.4.2	为控件添加属性和方法 .....	313
<b>第 11 章</b>	<b>ASP.NET 中使用 XML .....</b>	<b>316</b>
11.1	为什么要使用 XML .....	316
11.2	.NET 中的 XML 类 .....	317
11.3	使用 DataSet 对象操作 XML .....	318
11.3.1	从 XML 读取数据到 DataSet .....	318
11.3.2	从 DataSet 输出数据到 XML 文档 .....	319
11.3.3	DataSet 与 XmlDocument 一起使用 .....	321
11.4	使用 XML 控件显示 XML 数据 .....	323
11.4.1	直接用 XML 控件显示 XML 数据 .....	323
11.4.2	在 XML 控件中加载 DOM .....	325
11.5	使用 XslTransform 类转换 XML 数据 .....	328
11.6	XML 与数据库绑定综合应用 .....	329
<b>第 12 章</b>	<b>JavaScript 语言基础 .....</b>	<b>332</b>
12.1	JavaScript 语言概述 .....	332
12.1.1	什么是 JavaScript .....	332
12.1.2	JavaScript 的发展历程与特点 .....	333
12.1.3	JavaScript 和 Java 的区别 .....	334
12.1.4	编写第一个 JavaScript 程序 .....	336
12.2	JavaScript 语法基础 .....	338

12.2.1	数据类型与变量	338
12.2.2	表达式和运算符	342
12.3	JavaScript 程序结构	343
12.3.1	程序控制流程	344
12.3.2	JavaScript 函数与事件	348
12.4	JavaScript 中的 JSON	353
12.4.1	JSON 及其描述方法	353
12.4.2	JSON 与 XML 的比较	357
12.4.3	JSON 对象与字符方式转换	359
<b>第 13 章</b>	<b>JavaScript 对象简介</b>	<b>362</b>
13.1	JavaScript 对象的特征	362
13.2	JavaScript 对象的操作	363
13.2.1	JavaScript 对象的引用	363
13.2.2	JavaScript 对象的操作语句	364
13.3	JavaScript 对象与函数	366
13.3.1	JavaScript 对象的创建	366
13.3.2	JavaScript 的函数对象	368
13.4	JavaScript 内部对象	372
13.4.1	JavaScript 的 String(字符串)对象	372
13.4.2	JavaScript 的 Date(日期)对象	375
13.4.3	JavaScript 的 Boolean(逻辑)对象	377
13.4.4	JavaScript 的 Number(数值)对象	378
13.4.5	JavaScript 的 Array(数组)对象	379
13.4.6	JavaScript 的 Math 对象	382
13.4.7	JavaScript 中的系统函数	384
13.5	JavaScript HTML DOM 对象	385
13.5.1	什么是 HTML DOM 对象	385
13.5.2	Document 对象的属性与方法	387
13.5.3	Document 节点对象的访问	388
<b>第 14 章</b>	<b>基于 Ajax 的 Web 应用</b>	<b>393</b>
14.1	传统 Web 的弊病	393
14.2	Ajax 带来的变化及优势	394
14.2.1	Ajax 带来的变化	394
14.2.2	Ajax 技术的优势	395
14.3	Ajax 及技术实现	396
14.3.1	Ajax 简介	396
14.3.2	Ajax 工作原理	397

14.3.3	第一个 Ajax 程序 .....	398
14.3.4	Ajax 核心对象 XmlHttpRequest .....	400
14.3.5	一个典型 Ajax 调用过程 .....	404
14.4	Ajax 请求与响应实例 .....	405
14.5	基于 Ajax 的 Web 应用实例 .....	409
14.5.1	Ajax 访问数据库实例 .....	409
14.5.2	Ajax 访问 XML 数据文件实例 .....	411
14.5.3	Ajax ResponseXML 应用实例 .....	414
14.6	Ajax 与传统的 Web 应用与开发方式的差异 .....	416
14.6.1	应用模式的比较 .....	416
14.6.2	Ajax 编程的技术难点 .....	417
<b>第 15 章</b>	<b>ASP.NET Ajax 框架与组件 .....</b>	<b>419</b>
15.1	Ajax 框架与组件简介 .....	419
15.2	ASP.NET Ajax 的组成 .....	421
15.3	核心组件 ASP.NET 2.0 Ajax Extensions .....	425
15.3.1	使用 ScriptManager 控件 .....	426
15.3.2	使用 UpdatePanel 控件 .....	428
15.3.3	Ajax 与 WebServices 引用 .....	433
15.4	Ajax Library 类库 .....	438
15.5	应用工具包 AjaxControlToolkit .....	442
15.6	Ajax 编程的本质及开发途径 .....	445
	<b>参考文献 .....</b>	<b>447</b>

## 1.1 Web 应用及其特征

### 1.1.1 什么是 Web 应用

Web 应用(也称 Web 应用系统)是利用各种动态 Web 技术开发的基于浏览器/服务器(Browser/Server, B/S)模式的事务处理系统,其事务逻辑处理和数据的逻辑运算由 Web 服务器、数据库系统和客户端共同提供,用户借助浏览器在客户端运行,运行结果通过浏览器返回给用户。比如,电子商务、CRM 以及常见的网站系统(如电子政务网站、企业门户网站等)都是 Web 应用系统。

Web 应用程序首先是“应用程序”,与用标准的程序语言,如 C、C++、C#、PowerBuilder 和 VB 等编写出来的程序没有什么本质上的区别。然而 Web 应用程序又有自己的独特之处,就是它是基于 B/S 架构,能够很好地应用在广域网上,成为越来越多的企业的理想选择。

相对于早前广泛使用的客户端/服务器(Client/Server, C/S)应用程序体系结构, B/S 架构主要具有如下优点。

(1) 这种架构采用 Internet 上标准的通信协议作为客户端与服务器通信的协议,使位于 Internet 任意位置的用户都可以访问服务器,可实现数据的远程共享与实时处理。

(2) 将浏览器作为客户端的应用程序,以实现数据的显示。不再需要为客户端单独安装其他类型的应用程序。这样,在客户端只需要安装一套内置浏览器的操作系统,就可以实现服务器上数据的访问,简化了客户端的配置与管理,也便于应用系统的维护和更新。

常见的网络计算器、留言板、聊天室和论坛 BBS 等,都是 Web 应用程序,不过这些应用相对比较简单,而 Web 应用的真正核心和动力应该是对企业的业务进行处理。在企业级大型应用领域,Web 应用与目前广泛使用的桌面应用系统(Desktop Application System)在其可用性和用户体验方面还存在一定的差距。

桌面应用程序通常以 CD 为介质(有时候可从网站下载)并完全安装到本地的计算机上,即运行这些应用程序的代码在桌面计算机上。而 Web 应用程序运行在远程的 Web 服务器上,必须通过 Web 浏览器访问这种应用程序。比这些应用程序的运行代码放在何处更重要的是,应用程序如何运转以及如何与其进行交互。桌面应用程序响应速度快、使用方便、数据处理能力强,具有漂亮的用户界面和很强的动态交互性。

Web 应用程序是最新的潮流,它们提供了在桌面上不能实现的远程服务。其数据是集

中存放的,客户端发生的每一笔业务都直接进入中央数据库,可以很好地保证数据的一致性和实时性,并且不受时间和地域的限制。但是,伴随着 Web 的强大而出现的是服务器响应的延迟、页面刷新所带来的糟糕的用户体验、弱交互性以及功能的局限性。

交互性差、响应延迟和页面的频繁刷新是传统 Web 应用的主要障碍。不过,这种情况近几年正在发生改变,Ajax(Asynchronous JavaScript and XML,异步 JavaScript 和 XML)技术已经修复了传统 Web 应用中的这些棘手的问题,使得从桌面应用转向 Web 应用成为可能。Ajax 正尝试建立桌面应用程序的功能和交互性与不断更新的 Web 应用程序之间的桥梁。可以实现类似桌面应用程序中常见的用户界面以及良好的可用性与用户体验。

### 1.1.2 Web 网站与 Web 应用

在谈论 Web 应用程序时,很多人首先会想到 Web 网站(Web Site)。其实,Web 应用与 Web 网站是两个既有交叉也有区别的概念。Web 网站是伴随 Internet 技术而出现的一种由 Web 页面组成的、用于图文信息发布的网络媒体,每个 Web 页面是一个由各种文字、图片等信息组成的页面文档(Document),它是作为静态视图存储在服务器中,并通过浏览器下载到客户端显示,这些页面文档通过链接(Links)形成一个 Web 网站。早期的 Web 网站就是一个电子布告栏。为满足媒体的互动性,通过表单和简单的编程与后台的数据库连接,使得 Web 页面具有一定的交互功能。随着 Web 技术和开发工具的日益成熟,具有很少软件工程技术知识的人也可以很容易地使他们的网页具有交互性。不过,在 Web 上创建内容并不等于部署了 Web 应用。

Web 应用是一种经由 Internet 或 Intranet、以 Web 方式访问的应用程序,属于软件工程的范畴。它应具有以下显著的特点:它是作为单一实体、具有特定功能的计算机软件应用程序;它是通过网络浏览器来运行的,并通过服务器处理数据以提供信息服务。作为一个网站,用户关注的是它的内容、查询的效率和界面的效果;而对于一个 Web 应用,用户更关心的是其功能,也就是它所能提供的服务,并对其交互性有更高的要求。常见的最简单的 Web 应用有聊天室、留言板、博客或电子商务等。

当然,Web 应用与 Web 网站之间也有共同点:都是由 Web 页面组成的,并通过网络为用户提供信息服务;都要求有好的可用性和用户体验。

在传统 Web 中,用户唯一可能交互的形式是向 Web 表单输入数据并向服务器提交该表单,用户每次与服务器交互都获得一个新的文档页面。页面上的任何变动和操作都会导致整个页面的刷新,包括数据、样式和结构等所有的部分,例如更新 CRM 中的一条客户的信息。其主要问题是:页面的弱交互性、大量信息的网络传输引起页面响应的延迟以及页面的频繁刷新给用户带来糟糕的用户体验。

Ajax 技术的出现和使用从根本上改变了客户端与服务器的数据通信方式。Ajax 通过客户端技术、微请求(Micro Request)和异步通信等技术,使得原来以页面文档为单元的通信变为以页面中的数据对象为通信单元,显著减少了服务器的工作负荷和网络传输量。并通过使用异步方式与服务器通信,不需要中断用户的操作,从而使页面具有更迅速的响应能力和更强的交互能力,极大地提高了 Web 的数据处理能力。

目前 Ajax 已经成为 Web 应用开发的主流开发技术,并形成了 Ajax 技术应用的热潮,拓展了 Web 应用的领域与范围,催生了大量的新的 Web 应用。

## 1.2 Web 应用的相关技术

从软件工程角度看,Web 应用是具有特定功能的应用程序,但从用户观点看,Web 应用仍然是以页面的形式表现出来的。一个页面包括 3 种基本元素,即结构(Structure)、表现(Presentation)和行为(Behavior)。通常用结构化标记语言 HTML、XHTML 和 XML 来定义结构,用 CSS 设计样式,而用 JavaScript 等脚本来控制客户端行为。一个完整的 Web 应用还包括 Web 服务器和数据库服务。Web 服务器接收和处理客户端的请求,并提供计算服务;而数据库服务器则负责数据的存储、检索和控制等。服务器端的编程语言主要有 Java、C# 和 PHP 等。

由此可见,Web 应用涉及的技术非常多,而且这些技术是相互关联的,需要组合使用。下面对这些技术及作用做一个简要的介绍。

(1) HTML/XHTML: HTML(HyperText Markup Language,超文本标记语言)是一种用于制作超文本文档的简单标记语言。HTML 是最基本的 Web 页面设计语言,用 HTML 编写的超文本文档叫 HTML 文档。自 1990 年以来 HTML 就一直被用做 World Wide Web(WWW,万维网)的信息表示语言,它是所有 Web 网站和 Web 应用的基础。使用 HTML 描述的文件,需要通过浏览器来显示。所谓超文本,是因为它可以从一个文件跳到另一个文件,与世界各地主机中的文件连接,并且可以加入图片、声音、动画和视频等内容。HTML 文档的扩展名通常是 html、htm。XHTML(the eXtensible HyperText Markup Language,可扩展超文本标记语言)是对 HTML 的扩展。XHTML 中可以使用 XML 的语义功能,是一个基于 XML 的标记语言。本质上说,XHTML 是一个过渡技术,结合了部分 XML 的强大功能及大多数 HTML。

(2) CSS: CSS(Cascading Style Sheets,层叠样式表)负责控制页面的外观表现,如布局、颜色和字体等。CSS 提供了页面结构内容与表现形式的分离机制,大大简化了 HTML 页面的管理与维护工作,也提高了其开发的工作效率。CSS 可用于控制任何 HTML 和 XML 内容的表现形式。

(3) XML: XML(eXtensible Markup Language,可扩展标记语言)与 HTML 一样,都是出自 SGML(标准通用语言)。XML 是 Internet 环境中跨平台的、依赖于内容的技术,是当前处理结构化文档信息的主要工具。XML 也是一种简单的数据交换语言。XML 为分布式的、异构的数据交换提供了强大的功能,并且将数据本身和数据的表现分离。XML 的出现为程序能够自动地处理 Web 数据以及 Web 服务提供了一种公共基础。

(4) DOM: DOM(Document Object Model,文档对象模型)为 HTML、XML 等数据载体和信息载体在内存中的处理提供了一种基本的对象模型,即用对象来描述页面上的所有元素,是页面结构及组成元素的对象化表示方法,可提供连接到文件的结构、格式、事件等。DOM 不依赖于任何程序设计语言和页面描述语言,它为处理 HTML 和 XML 数据提供了一种标准的、独立的接口。

(5) JavaScript: JavaScript 是为适应动态网页的交互性需要而诞生的一种脚本(Script)语言。JavaScript 是由 Netscape 公司开发的一种脚本语言,借助 JavaScript 等脚本语言,可以在 Web 页面实现客户端与服务器端的动态交互,JavaScript 的出现使得网页和用



户之间实现了实时的动态交互。JavaScript 短小精悍,又是在客户机上执行,大大提高了网页的浏览速度和交互能力。同时它又是专门为制作 Web 网页而量身定做的一种简单的编程语言,如今 JavaScript 越来越广泛地使用于 Web 开发上。JavaScript 可以被应用为一种客户端或者服务器端技术。当应用在客户端时,JavaScript 被嵌入到发送给 Web 浏览器的 HTML 中,并且直接被浏览器解释。JavaScript 也可以被应用为一种创建动态内容时使用的服务器端技术,其他的服务器端技术(如微软的 ASP.NET)对 JavaScript 提供相应的支持。

(6) Java: Java 是 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言(Java 语言)和 Java 平台的总称。它最初的名字是 Oak,被设计用来在嵌入式系统中运行,后来被用做了基于 Web 的技术。Java 平台由 Java 虚拟机和 Java 应用编程接口(Java API)构成。Java API 为 Java 应用提供了一个独立于操作系统的标准接口,使得 Java 具备了平台无关性。跨平台、动态的 Web、Internet 计算是 Java 的主要特征,它经常被 Web 服务器使用以发布复杂的交互式应用。

(7) .NET: .NET 是一种开发平台,而不是一种语言。微软于 2002 年引入了 .NET,并将其作为包括 Web 应用在内的许多应用程序的开发平台。它和 Java 具有类似性,因为其源代码也被编译为称为公共中间语言(Common Intermediate Language, CIL)的中间代码,然后该中间代码再由 .Net Framework 翻译成机器码并执行。可以使用具有语言无关性的 .NET 来设计 Web 应用,而应用程序则可以使用任意与 .NET 兼容的语言来编写。

.NET 平台主要包括 4 个部分的内容: 低层操作系统、.NET 企业服务器、.NET 框架和集成开发工具 Visual Studio .NET。在 Visual Studio .NET 中可以用 C#、C++、Basic、J# 语言开发。可以开发桌面应用程序、Web 应用程序、智能设备应用程序等。

在 .NET 平台上,C# 语言是开发的主要语言。C# 是由 Microsoft 开发的一种新型编程语言,它是从 C 和 C++ 中派生出来的,因此具有 C++ 的功能。对于 Web 开发而言,C# 类似 Java 开发语言。所以,C# 是开发 .NET 框架应用程序的最好语言。.NET 平台将 C# 作为其固有语言。

此外,当前和未来的 Web 开发不得不提到近几年最流行的一个词语 Ajax。Ajax 是当前非常流行的 Web 开发技术。事实上,Ajax 并不是什么新的技术,只不过是 JavaScript 一些相对比较高级的特性的综合应用。一般来说,Ajax 所涉及的 JavaScript 特性主要包括:

- 通过 JavaScript 的 XMLHttpRequest 对象实现异步请求与调用。
- 通过 JavaScript 对 XML 进行解析和处理。
- 通过 JavaScript 操作 DOM 来实现客户端的动态显示。

在传统 Web 中,用户唯一可能交互的形式是向 Web 表单输入数据并向服务器提交该表单。用户每次与服务器交互都获得一个新的文档页面。页面上的任何变动和操作都会导致整个页面的刷新,包括数据、样式和结构等所有的部分,例如更新 CRM 中的某一个客户的信息会导致页面上所有客户信息的重新加载。

Ajax 改变了传统 Web 应用的模式,Ajax 技术既是对传统 Web 应用的完善,也是对传统 Web 应用的革命。Ajax 不仅可以改善用户体验,还可以简化 Web 开发,通过将页面高度模块化,数据与表现分离,从而可以使服务器端和客户端都可以很好地解耦,降低开发的复杂度。