



DVD语音视频讲解光盘

120个网页实例及程序源代码

60个语音视频讲解实现及环境搭建过程

附赠7种网络和应用工具软件

Ajax

完全学习手册

张银鹤 梁文新 李新磊 等编著



本书内容特色

Ajax核心技术以及框架应用

涵盖从事Ajax开发所要掌握的知识

IP查询城市、在线裁切图片、RSS阅读器

在线相册系统展现Web 2.0下使用Ajax技术的用户体验

窗内网 (www.itzcn.com) 提供技术支持

清华大学出版社





DVD语音视频讲解光盘

120个网页实例及程序源代码

60个语音视频讲解实现及环境搭建过程

附赠7种网络和应用工具软件

Ajax

完全学习手册

张银鹤 梁文新 李新磊 等编著



本书内容特色

Ajax核心技术以及框架应用

涵盖从事Ajax开发所要掌握的知识

IP查询城市、在线裁切图片、RSS阅读器

在线相册系统展现Web 2.0下使用Ajax技术的用户体验

窗内网 (www.itzcn.com) 提供技术支持

清华大学出版社

北 京

内 容 简 介

本书全面介绍 Ajax 动态网站开发知识。书中 Ajax 基础篇介绍 Ajax 概念、在 ASP.NET 中的常见开发模式；ASP.NET Ajax 应用篇介绍 Ajax 技术在 ASP.NET 中的简单应用；ASP.NET Ajax 框架篇介绍使用 ASP.NET Ajax 框架实现各种特效；Ajax 高级技术篇介绍如何增强 Ajax 应用的性能，包括优化 Ajax 应用以确保程序质量、应用 Ajax 设计模式、Web 安全问题以及保护数据等方面；综合实例篇实现一个在线相册系统。本书配套光盘内容为本书的源代码。

本书适合于中、高级动态网站开发人员，特别适合于有编程基础，希望全面学习 Ajax 技术，提高实际应用能力的读者群体。

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

Ajax 完全学习手册 / 张银鹤等编著. —北京：清华大学出版社，2009.4
ISBN 978-7-302-19217-6

I. A… II. 张… III. 计算机网络—程序设计 IV. TP393.09

中国版本图书馆 CIP 数据核字 (2009) 第 001032 号

责任编辑：夏兆彦

责任校对：徐俊伟

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：北京市世界知识印刷厂

装 订 者：三河市李旗庄少明装订厂

经 销：全国新华书店

开 本：190×260 印 张：34.5 字 数：857 千字

附光盘 1 张

版 次：2009 年 4 月第 1 版 印 次：2009 年 4 月第 1 次印刷

印 数：1~4000

定 价：59.50 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系
调换。联系电话：(010)62770177 转 3103 产品编号：029957-01

前 言

Ajax (异步 JavaScript 与 XML) 是一种使用客户端脚本与 Web 服务器交换数据的 Web 应用开发方法。Web 页面不用打断交互流程进行重新加载, 就可以动态地更新。使用 Ajax, 可以创建接近本地桌面应用的、直接的、高可用的、更丰富的、更动态的 Web 用户接口界面。

Ajax 技术的广泛应用, 已被证明该技术受市场热烈欢迎, 同时也证明技术的正确性。每一个 Ajax 技术的使用者都成为了胜利者, 包括 Google、yahoo、Amazon 和微软等。是 Google 地图吸引了 Web 开发人员的目光, 人们由此揭开了 Ajax 神秘的面纱。

1. 本书主要内容

第一篇为 Ajax 基础篇。本篇首先向读者阐述了 Ajax 的概念、Ajax 的优势、Ajax 运行机制以及 Ajax 在 ASP.NET 中的常见开发模式。然后, 对 Ajax 的核心组成部分进行了详细介绍, 包括 CSS、JavaScript、XML、XMLHttpRequest 及 DOM 等。

第二篇为 ASP.NET Ajax 应用篇, 主要介绍 Ajax 技术在 ASP.NET 中的简单应用。本篇首先从 ASP.NET 的概念入手, 介绍 ASP.NET 环境配置方法、内置对象及访问外部存储介质 (文件、XML 和数据库) 的方法。接着介绍服务器端对客户端发送的请求进行处理并返回相应数据的过程, 最后介绍各种经典案例的实现, 包括 IP 查询城市、在线裁切图片、自动完成和保存草稿以及 RSS 阅读器。

第三篇为 ASP.NET Ajax 框架篇, 主要介绍使用 ASP.NET Ajax 框架实现各种特效。本篇首先对 Ajax 框架进行概述并列举了常用的各种框架。接着以 ASP.NET Ajax 框架为例详细讲述配置、使用以及实现过程。本篇中介绍的特效包括可折叠面板、水印文本框、自动完成、级联菜单、可拖曳面板、列表搜索、评级和密码强度等。

第四篇为 Ajax 高级技术篇。本篇重点向读者介绍如何增强 Ajax 应用的性能, 包括优化 Ajax 应用以确保程序质量、应用 Ajax 设计模式、Web 安全问题以及保护数据等方面。

第五篇是综合实例篇。本篇以实现一个在线相册系统的过程为例进行讲解, 让读者全面了解并熟悉如何将 Ajax 应用到实际项目的过程。其中包括很多实现技巧, 如自动播放和照片显示等功能。

2. 本书主要特点

本书全面介绍了 Ajax 动态网站开发和应用知识。本书具有下面的特点。

- **内容全面** 本书是一本大全性质的 Ajax 编程图书, 突出介绍 Ajax 动态网站开发相关知识。读者学习本书之后, 可以全面掌握 Ajax 动态网站开发实践知识。
- **实例丰富** 全书每个知识点的讲解都配有大量可实际运行的实例。读者可以边学习边实践, 快速、全面地掌握 Ajax 的开发方法和技巧。书中最后一篇还提供了一个典型开发案例, 覆盖了利用 Ajax 开发和部署网站的全部过程。

本书配套光盘提供了本书所有源代码 (包括一个完整的应用程序) 和 ASP 软件配置和使

用教学视频。读者只要按照书中的范例上机练习，举一反三，就可以根据自己的需要开发出功能强大的 Web 动态网站。

3. 本书读者对象

本书结构清晰，语言通俗易懂，适合于中、高级动态网站开发人员，特别适合于有编程基础，希望全面学习 Ajax 技术，提高实际应用能力的读者群体。本书也可作为高等院校的教学用书和相关培训机构的培训教材。

除了封面署名人员之外，参与本书编写的还有于永军、张秋香、李乃文、张仕禹、夏小军、赵振江、李振山、李文才、吴越胜、李海庆、何永国、李海峰、陶丽、吴俊海、安征、张巍屹、崔群法、王咏梅、康显丽、辛爱军、牛小平、贾栓稳、王立新、苏静、赵元庆、郭磊、徐铭、李大庆、王蕾、张勇、郝安林、郭新志、牛丽平、唐守国等。

在编写过程中难免会有疏漏之处，欢迎读者与我们联系，请给予指正，特此感谢！

编 者

2008 年 12 月

目 录

第一篇 Ajax 基础篇

第 1 章 Ajax 概述 1

- 1.1 传统 Web 应用解决方案 1
- 1.2 新的 Web 开发模式——Ajax 3
 - 1.2.1 初识 Ajax 3
 - 1.2.2 Ajax 开发模式 6
- 1.3 Ajax 运行机制 7
- 1.4 第一个 Ajax 示例 9

第 2 章 CSS 样式 16

- 2.1 CSS 概述 16
 - 2.1.1 CSS 简介 16
 - 2.1.2 CSS 基础语法 17
 - 2.1.3 CSS 颜色和单位 22
 - 2.1.4 第一个示例 25
- 2.2 引用 CSS 样式 26
 - 2.2.1 内联样式表 26
 - 2.2.2 嵌入样式表 27
 - 2.2.3 外部样式表 30
- 2.3 颜色及背景 31
 - 2.3.1 color 属性 32
 - 2.3.2 background-color 属性 33
 - 2.3.3 background-image 属性 33
 - 2.3.4 background-repeat 属性 34
 - 2.3.5 background-position 属性 35
 - 2.3.6 background-attachment 属性 36
 - 2.3.7 background 属性 37
- 2.4 字体属性 38
 - 2.4.1 font-family 属性 38
 - 2.4.2 font-weight 属性 39
 - 2.4.3 font-size 属性 39
 - 2.4.4 font-style 属性 41

2.4.5 font-variant 属性 41

2.4.6 font 属性 42

2.5 文本属性 43

2.5.1 text-indent 属性 43

2.5.2 text-align 属性 44

2.5.3 white-space 属性 45

2.5.4 line-height 属性 46

2.5.5 vertical-align 属性 47

2.5.6 text-transform 属性 48

2.5.7 text-decoration 属性 49

2.5.8 word-spacing 属性和 letter-spacing
属性 50

2.6 边框属性 50

2.6.1 border-style 属性 51

2.6.2 border-width 属性 52

2.6.3 border-color 属性 53

2.6.4 border 属性 54

2.7 定位与布局 54

2.7.1 定位属性 54

2.7.2 边偏移属性 55

2.7.3 position 属性 56

2.7.4 overflow 属性 60

2.7.5 float 属性 62

2.7.6 visibility 属性和 display 属性 64

第 3 章 JavaScript 67

3.1 JavaScript 语言基础 67

3.1.1 JavaScript 简介 67

3.1.2 编写 JavaScript 69

3.2 程序设计基础 70

3.2.1 基本语法 70

3.2.2 运算符与表达式 72

3.2.3	流程控制语句	75
3.2.4	异常处理	84
3.2.5	对话框	85
3.3	内置对象	89
3.3.1	JavaScript 对象概述	89
3.3.2	String 对象	91
3.3.3	Math 对象	96
3.3.4	Date 对象	102
3.3.5	Array 对象	105
3.4	浏览器对象	109
3.5	自定义对象	111
3.5.1	基于 Object 创建对象	111
3.5.2	基于 prototype 创建对象	113
第 4 章	XML	115
4.1	XML 概述	115
4.1.1	XML 简介	115
4.1.2	XML 语法	118
4.1.3	XML 标记和元素	120
4.1.4	XML 属性	122
4.1.5	特殊字符及 CDATA 区段	123
4.2	DTD	125
4.2.1	DTD 简介	125
4.2.2	内部和外部 DTD	126
4.2.3	声明 DTD	129
4.2.4	DTD 实体	133
4.3	XSLT	138
4.3.1	XSLT 简介	138
4.3.2	XSLT 文档	139
4.3.3	XSLT 模板语法	140
4.3.4	XSLT 元素	144

第 5 章 XMLHttpRequest 148

5.1	XMLHttpRequest 简介	148
5.2	XMLHttpRequest 核心	149
5.2.1	XMLHttpRequest 运行周期	149
5.2.2	XMLHttpRequest 对象属性	151
5.2.3	XMLHttpRequest 对象方法	153
5.3	使用 XMLHttpRequest	155
5.3.1	创建 XMLHttpRequest 对象	155
5.3.2	发送请求	156
5.3.3	处理回调函数	158
5.4	XMLHttpRequest 简单应用	160
5.4.1	用户注册实例	160
5.4.2	处理 XML 请求	163
5.4.3	动态列表	167

第 6 章 DOM 170

6.1	DOM 概述	170
6.1.1	DOM 简介	170
6.1.2	DOM 与 HTML	171
6.1.3	DOM 与 XML	173
6.2	DOM 核心	174
6.2.1	DOM 核心接口	174
6.2.2	DOM 核心对象	176
6.3	DOM API	177
6.3.1	Document	177
6.3.2	Node 和 NodeList	179
6.3.3	Element	182
6.3.4	Text 和 Attr	184
6.4	DOM 操作	186
6.4.1	操作 HTML 文档	186
6.4.2	操作 XML 文档	190

第二篇 ASP.NET Ajax 应用篇

第 7 章 Ajax 服务器端编程 195

7.1	ASP.NET 概述	195
7.1.1	ASP.NET 简介	195
7.1.2	配置 ASP.NET 环境	197

7.2	ASP.NET 开发技术	202
7.2.1	内置对象	202
7.2.2	数据库	207
7.3	实例应用	210
7.3.1	使用内置对象	210

7.3.2 访问文件	216
7.3.3 访问 XML	219
7.3.4 访问数据库	225
7.3.5 访问 Web 服务	230
第 8 章 与客户端简单交互	233
8.1 HTTP 概述	233
8.1.1 请求	233
8.1.2 响应	235
8.2 发送 HTTP 请求	236
8.2.1 普通文本请求	236
8.2.2 带格式请求	239
8.3 处理服务器响应	243

8.3.1 普通文本响应	243
8.3.2 带格式响应	246
8.4 综合实例	254

第 9 章 Ajax 基本技术实现

9.1 通过 IP 查询城市	267
9.2 在线裁切图片	272
9.3 弹出栏目提示	278
9.4 搜索栏自动完成	282
9.5 自动保存草稿	285
9.6 在线设计名片	291
9.7 RSS 阅读器	296

第三篇 ASP.NET Ajax 框架篇

第 10 章 Ajax 框架

10.1 Ajax 框架概述	303
10.2 ASP.NET Ajax 框架	305
10.2.1 ASP.NET Ajax 框架简介	305
10.2.2 配置 ASP.NET Ajax 环境	306
10.3 ASP.NET Ajax 核心控件	309
10.3.1 ScriptManager 控件	309
10.3.2 ScriptManagerProxy 控件	314
10.3.3 Timer 控件	316
10.3.4 UpdatePanel 控件	319
10.3.5 UpdateProgress 控件	322

第 11 章 ASP.NET Ajax 实用技能一

11.1 Accordion 控件	324
11.1.1 Accordion 控件简介	325
11.1.2 Accordion 控件实例	327
11.2 AlwaysVisibleControlExtender 控件	330
11.2.1 AlwaysVisibleControlExtender 控件简介	330
11.2.2 AlwaysVisibleControlExtender 控件实例	330
11.3 AnimationExtender 控件	333
11.3.1 AnimationExtender 控件简介	333

11.3.2 AnimationExtender 控件实例	335
11.4 AutoCompleteExtender 控件	339
11.4.1 AutoCompleteExtender 控件简介	339
11.4.2 AutoCompleteExtender 控件实例	340
11.5 CalendarExtender 控件	342
11.5.1 CalendarExtender 控件简介	342
11.5.2 CalendarExtender 控件实例	343
11.6 CascadingDropDown 控件	345
11.6.1 CascadingDropDown 控件简介	345
11.6.2 CascadingDropDown 控件实例	346
11.7 CollapsiblePanelExtender 控件	350
11.7.1 CollapsiblePanelExtender 控件简介	351
11.7.2 CollapsiblePanelExtender 控件实例	352
11.8 ConfirmButtonExtender 控件	353
11.8.1 ConfirmButtonExtender 控件简介	353
11.8.2 ConfirmButtonExtender 控件实例	354

第 12 章 ASP.NET Ajax 实用技能二……

12.6.2 NoBot 控件实例	404	12.12.1 ReorderList 控件简介	431
12.7 NumericUpDownExtender 控件	408	12.12.2 ReorderList 控件实例	434
12.7.1 NumericUpDownExtender 控件简介	408	12.13 ResizableControlExtender 控件	438
12.7.2 NumericUpDownExtender 控件实例	409	12.13.1 ResizableControlExtender 控件简介	438
12.8 PagingBulletedListExtender 控件	412	12.13.2 ResizableControlExtender 控件实例	439
12.8.1 PagingBulletedListExtender 控件简介	412	12.14 RoundedCornersExtender 控件	440
12.8.2 PagingBulletedListExtender 控件实例	413	12.14.1 RoundedCornersExtender 控件简介	441
12.9 PasswordStrength 控件	417	12.14.2 RoundedCornersExtender 控件实例	441
12.9.1 PasswordStrength 控件简介	417	12.15 SliderExtender 控件	444
12.9.2 PasswordStrength 控件实例	418	12.15.1 SliderExtender 控件简介	444
12.10 PopupControlExtender 控件	420	12.15.2 SliderExtender 控件实例	445
12.10.1 PopupControlExtender 控件简介	421	12.16 SlideShowExtender 控件	446
12.10.2 PopupControlExtender 控件实例	421	12.16.1 SlideShowExtender 控件简介	447
12.11 Rating 控件	423	12.16.2 SlideShowExtender 控件实例	447
12.11.1 Rating 控件简介	423	12.17 TabContainer 控件	449
12.11.2 Rating 控件实例	424	12.17.1 TabContainer 控件简介	449
12.12 ReorderList 控件	431	12.17.2 TabContainer 控件实例	450

第四篇 Ajax 高级技术篇

第 13 章 增强 Ajax 性能	452	13.3.3 Observer 模式	467
13.1 优化 Ajax	452	13.3.4 Singleton 模式	469
13.1.1 确保程序质量	453	13.3.5 Command 模式	471
13.1.2 处理请求的响应	454	13.4 Web 应用的安全问题	475
13.1.3 通知框架	457	13.5 保护数据	477
13.2 Ajax 设计模式	459	13.5.1 使用安全 HTTP	478
13.2.1 设计模式简介	459	13.5.2 JavaScript 加密数据	479
13.2.2 设计模式原则	460	13.6 测试 Ajax 性能	480
13.2.3 基本设计模式	461	13.6.1 使用 Venkman 性能分析器	480
13.3 常用 Ajax 设计模式	463	13.6.2 JavaScript 执行速度	482
13.3.1 Adapter 模式	464	13.6.3 JavaScript 内存情况	485
13.3.2 Facade 模式	466		

第五篇 综合实例篇

第 14 章 在线相册系统 489

14.1 系统概述 489

14.1.1 需求分析 490

14.1.2 结构设计 490

14.2 通用类及代码设计 492

14.2.1 设计表 492

14.2.2 设计数据库类 494

14.2.3 文件处理类 498

14.2.4 通用代码设计 501

14.3 系统前台 505

14.3.1 系统首页 505

14.3.2 单个专辑内照片显示页面 510

14.3.3 单张照片显示页面 522

14.4 系统管理后台 525

14.4.1 用户登录 526

14.4.2 管理后台首页面 528

14.4.3 上传文件页面 531

14.4.4 添加专辑页面 532

14.4.5 编辑专辑页面 535

14.4.6 删除专辑 537

14.4.7 添加照片页面 538

第一篇 Ajax 基础篇

第 1 章 Ajax 概述



内容摘要 | Abstract

Ajax 全称 Asynchronous JavaScript and XML，中文含义为“异步 JavaScript 和 XML”。它是 Web 2.0 技术的核心，由多种技术组合而成。使用 Ajax 技术不必刷新整个页面，只需对页面的局部进行更新，可以节省网络带宽，提高网页加载速度，从而缩短用户等待时间，改善用户操作体验。

本章将对 Ajax 进行全面介绍，使读者对传统 Web 应用解决方案、Ajax 的开发模式、Ajax 的运行机制有一个概要的了解，为以后章节的学习打下基础。本章最后将引导读者开发一个简单的 Ajax 案例。



学习目标 | Objective

- 理解传统 Web 应用解决方案
- 理解 Ajax 技术
- 理解 Ajax 开发模式
- 掌握理解 Ajax 的运行机制
- 理解会员验证方法
- 熟悉 Ajax 应用开发过程

1.1 传统 Web 应用解决方案

在传统 Web 应用解决方案中，如果用户使用浏览器浏览网页，当页面刷新很慢的时候，用户的浏览器在干什么？屏幕显示什么内容？通常情况下，用户的浏览器在等待刷新，而屏幕上则是一片空白，用户只能在屏幕前苦苦地等待浏览器的响应。开发人员为了克服这种尴尬的局面，不得不在每一个可能需要长时间等待响应的页面上增加一个 DIV 标记，在需要时显示类似“系统正在处理您的请求……”等信息以提示用户，从而改善用户的操作体验。

为什么会出现上述情况？因为传统的 Web 应用采用同步交互方式。这种方式下用户首先向 HTTP 服务器发送一个请求，服务器接到请求后开始执行用户请求的操作，最后将执行结果返回给用户浏览器，如返回一个 HTML 页面。这是一种不连贯的用户体验，服务器在处理请求的

时候, 用户多数时间处于等待状态, 屏幕上通常一片空白。图 1-1 和图 1-2 所示分别为传统 Web 应用程序的结构和模型。

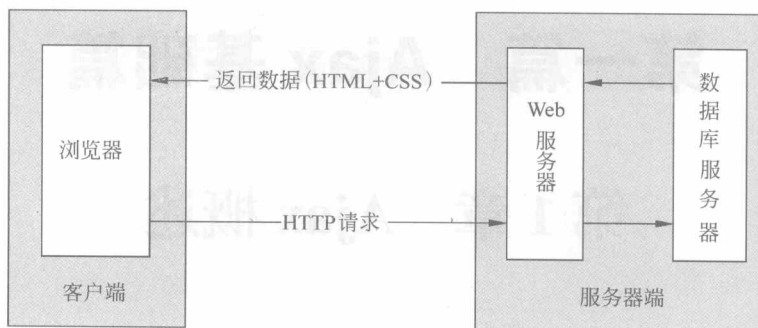


图 1-1 传统 Web 应用程序的结构

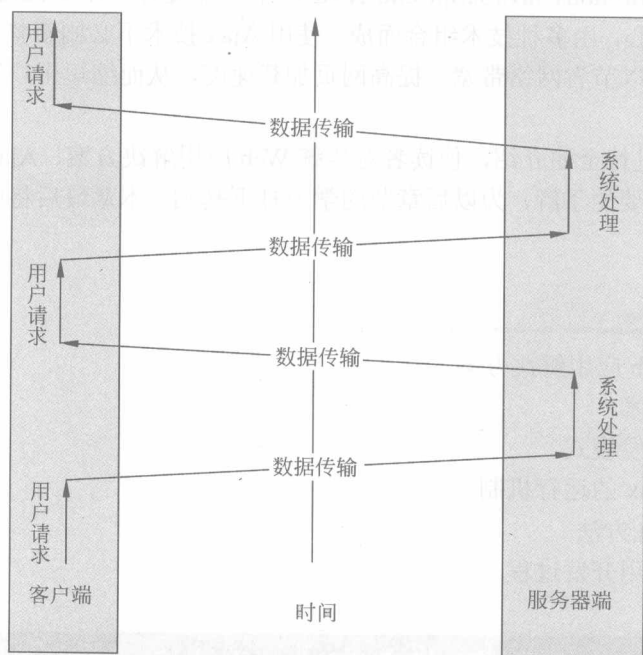


图 1-2 传统 Web 应用程序的模型

自从采用超文本作为 Web 传输和显示方式之后, Web 开发人员就一直使用这样的方式。当负载比较小的时候, 这种方式并没有体现出任何问题。可是当负载比较大时, 响应时间可能要很长, 一分钟、两分钟, 甚至数分钟的等待是用户不可忍受的。而且, 如果响应时间过长, 超过了页面响应时间, 那么服务器会返回页面不可用的信息。另外, 有时, 用户只是想改变页面中某一部分的数据, 而并非要重新加载整个页面。当软件设计越来越讲究人性化的时候, 这么糟糕的用户体验与这种原则背道而驰。为何总是要让用户等待服务器返回数据? 至少, 应该通过一定的方法来减少用户等待的时间。现在, 除了程序设计、编码优化和服务器调优之外, 还可以采用 Ajax 技术。

1.2 新的 Web 开发模式——Ajax

Ajax 用来描述一组技术, 主要包括: 客户端脚本语言 JavaScript、异步数据获取技术 XMLHttpRequest、数据互换和操作技术 XML 和 XSLT、动态显示和交互技术 DOM 及基于标准的表示技术 XHTML 和 CSS 等。Ajax 极大地发掘了 Web 浏览器的潜力, 开启了大量新的可能性, 从而有效地改善了用户操作体验。

1.2.1 初识 Ajax

在 Ajax 之前, Web 站点强制用户进入提交、等待、重新显示返回结果的过程。用户的操作总是与服务器的处理同步。Ajax 提供与服务器异步通信的能力, 从而使用户从请求-响应的循环中解脱出来。借助于 Ajax, 可以在用户单击按钮时, 使用 JavaScript 和 DHTML 立即更新用户页面, 并向服务器发出异步请求执行更新或查询数据库操作。当请求返回时, 就可以使用 JavaScript 和 CSS 来相应地更新页面中的某一部分, 而不是刷新整个页面。最重要的是, 用户甚至不知道浏览器正在与服务器通信, Web 站点看起来像是即时作出响应的。

1. Ajax 介绍

传统的 Web 应用允许用户填写表单(Form), 当提交表单时就向 Web 服务器发送一个请求。服务器接收并处理传来的表单, 然后返回一个新的网页。这种做法浪费了许多带宽, 因为前后两个页面中的大部分 HTML 代码往往是相同的。由于每次应用的交互都需要向服务器发送请求, 应用的响应时间就取决于服务器的响应时间。这导致用户界面的响应比本地应用要慢得多。

与此不同, Ajax 应用可以仅向服务器发送并取回必需的数据。它使用 SOAP 或其他一些基于 XML 的 Web Service 接口, 并在客户端采用 JavaScript 处理来自服务器的响应。因为服务器和浏览器之间交换的数据大量减少, 从而使得 Web 应用响应更快。同时, 很多处理工作可以在发出请求的客户端机器上完成, Web 服务器的处理量减少了, 这样能够大大缩短 Web 服务器的响应时间。总的来说, Ajax 应用程序具有以下特点。

- 通过异步模式, 提升用户体验。
- 优化浏览器和服务器之间的数据传输, 减少不必要的往返数据, 减少了带宽占用量。
- Ajax 引擎在客户端运行, 承担了一部分原本由服务器承担的工作, 从而减少了大用户量下的服务器负载。

Ajax 技术应用是受制于浏览器的。如果大量用户还在使用比较旧的浏览器, 如 IE 5、Mozilla 1.0 或 Safari 1.2 之前的版本, Ajax 技术的使用就失去了效果。另外, 不同浏览器上的 JavaScript 和 DOM 实现存在差异。开发人员在创建 XMLHttpRequest 对象的实例时一定要注意到这一点。这些内容将在后面进行详细介绍。

2. Ajax 与传统 Web 应用的区别

Ajax 技术是 Web 2.0 技术的重要组成部分。Ajax 技术不同于传统的 Web 技术, 是对传统 Web 技术的一种改良和发展。引入 Ajax 技术后, 不仅改进了 Web 应用的性能, 也改善了用户

的体验。下面就从几个方面介绍一下 Ajax 与传统 Web 应用之间的不同。

□ 用户体验

这是 Ajax 技术最大的改善之处。对于传统的 Web 应用，用户只能发送独占式请求，一旦请求发送出去，页面就处于等待状态，等待服务器完成响应。在服务器响应完成之前，客户端的浏览器只能是一片空白。而 Ajax 技术则完全不同。它允许客户端采用异步的方法发送请求，请求的发送完全不会阻塞当前的浏览器线程，浏览器可以继续下一步操作，如继续浏览，甚至再次发送异步请求。对于用户体验，Ajax 带来了重大的改善，它让用户不会处于等待状态，用户会感觉自己一直与应用处于交互状态。

□ 响应速度

就响应速度而言，大多数人会认为 Ajax 应用的速度比传统 Web 应用要快，实际上这种说法并不完全正确。基于 Ajax 的应用需要增加大量 JavaScript 代码，增加大量 JavaScript 代码后的 Web 页面在第一次加载时速度将比传统 Web 页面慢（必须下载大量的 JavaScript 代码）。一旦进入该页后，响应速度便会明显提高，无须频繁地在各页面之间跳转，从服务器获得的仅是必须改变的数据，减少了冗余数据的下载，从而大幅度提高响应速度。也许有人会认为 Ajax 包含的大量 JavaScript 代码会占用用户的大量带宽，这是不正确的，因为 Ajax 应用下载一次页面后，便可以多次重复使用。一次下载的 JavaScript 代码量虽然增大，但从长时间来看，传统 Web 应用需要多次下载 Web 页面，需要的带宽更大。

□ 应用架构

传统 Web 应用主要由 3 层组成，而增加 Ajax 技术的 Web 应用将在传统的 Web 应用上额外增加一个 Ajax 引擎，其实质就是一层 JavaScript 代码。这些 JavaScript 代码可以在客户端保存用户状态而无须使用 Session，能将控制器的部分功能转移到客户端页面，但这必然会导致安全性等方面的问题，需要开发者认真对待。

□ 开发的代码量

Ajax 技术的大部分功能都依赖于 JavaScript 语言实现，大量的 JavaScript 代码严重降低了程序员的开发效率。JavaScript 本身不是面向对象的编程语言，这严重限制了 JavaScript 代码的可重用性。JavaScript 代码并没有一个完善的调试工具，这无形中也加重了程序员的负担。

□ 服务器的负担

传统的观点认为 Ajax 技术减轻了服务器的负担，因为服务器只需要生成客户端必须更新的数据。这种说法在某些场合下也许正确，但实际的情形则是：大量使用 Ajax 技术的 Web 应用将导致服务器的负担大大加重，而绝不是减轻。因为 Ajax 技术往往比传统 Web 应用需要发送更多的请求，如用户注册时的用户名验证。传统 Web 应用无需发送任何请求，只要等待用户输入即可，而 Ajax 技术则是用户每输入一个字符，应用都将向服务器发送一次请求。

Ajax 技术的优势是相当明显的，但是应该在什么时候使用 Ajax 技术这是一个很重要的问题。如果过度使用 Ajax，那么就会有数千行 JavaScript 代码在客户端的浏览器上运行，可能会让用户感觉速度很慢。如果这些脚本编写不当，就会很快失去控制，特别是在通信量很大的情况下。所以，Ajax 技术的使用有一定的范围，比如在改善用户操作体验等方面。

3. Ajax 的应用

Ajax 的主要特点在于异步交互，更新 Web 页面的局部信息。因此，它比较适用于交互

较多、读取数据频繁，而数据传输量又比较小的 Web 应用。表 1-1 所示为 Ajax 的几种应用情况。

表 1-1 Ajax 的应用情况

应用情况	说明
基于表单的简单用户交互	如用户注册验证、数据格式验证等。如果采取整页回发 (PostBack) 到服务器做法，不仅传输数据量大、服务器负担重、响应时间长，而且会导致用户体验很差。在验证结束后，由于某些数据错误返回注册页面时，先前输入的数据都已不存在了，必须重新填写。而采用 Ajax 技术后，这些任务在用户填写其他信息时，已有 XMLHttpRequest 对象异步完成，极大地改善用户操作体验
时时更新的页面信息	如聊天室信息、在线统计、股票的涨跌等。这类系统都需要实时地反映数据的变化。采用 Ajax 技术能定时异步访问服务器，可以获得最新信息并将其显示给用户，而且可以避免整个页面的刷新
菜单导航	如多级联动菜单、树状导航等。可以采用 Ajax 技术来实现按需读取数据的功能，这可以避免每次变动都需要整页回发 (PostBack) 到服务器，从而节省带宽资源，提高响应速度，也减少显示所有数据时所消耗的带宽资源
评论、选择投票	这几种情况传输的数据量非常小，因而没有必要将整个页面回发 (PostBack) 到服务器。采用 Ajax 技术，用户在执行完相关操作后，将异步与服务器进行自动交互，而用户同时可以继续执行其他操作

大家或多或少使用过 Google 搜索服务，它就是使用 Ajax 技术实现交互的一个很好的示例。当用户键入字符时，Google Suggest 可以为用户提供与输入字符相符的提示，帮助用户完成想要输入的搜索字符串。图 1-3 所示为输入字符串“科技”后的效果。

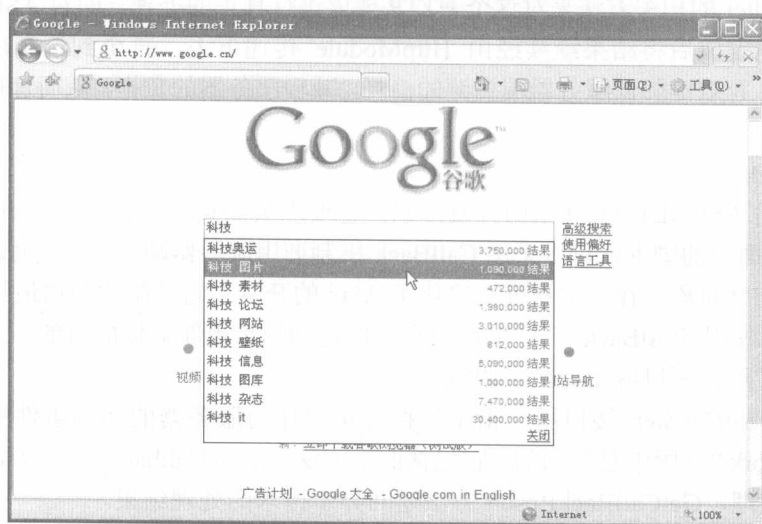


图 1-3 Google Suggest 的 Ajax 使用

从图 1-3 所示可知，Google Suggest 已经显示了建议性的内容，极大地方便了用户的搜索，从而改善了用户的搜索体验。

1.2.2 Ajax 开发模式

国内使用 Ajax 技术进行 Web 开发还是最近刚流行起来的, Ajax 技术还处于发展阶段, 并没有形成一个非常统一的开发模式。这些开发模式, 针对不同的 Ajax 项目, 在具体实现上的差别是比较大的。下面介绍几种常见的 Ajax 开发模式。

□ XMLHTTP+WebForm

XMLHTTP+WebForm 是使用 Ajax 技术进行 Web 应用开发的最基本方法。首先, 开发人员使用 JavaScript 语言编写代码操作 XMLHTTP 对象, 向服务器端的某个 ASP.NET 页面发出异步请求。服务器端的 ASP.NET 页面接收请求, 并根据请求进行相应的处理, 处理完成后返回相应的执行结果给 XMLHTTP 对象。最后, 开发人员再使用 JavaScript 语言编写代码把返回的结果显示在客户端的浏览器上。

□ XMLHTTP+HttpHandler

XMLHTTP+HttpHandler 与 XMLHTTP+WebForm 相比在客户端的实现上并没有太大变化, 也是使用 JavaScript 语言编写代码操作 XMLHTTP 对象, 但是服务器端改用 HttpHandler 来接收和处理异步请求。

HttpHandler 是 HTTP 请求的真正处理中心。正是在这个 HttpHandler 容器中, ASP.NET Framework 才真正对客户端请求的服务器页面进行编译和执行, 并将处理后的信息附加在 HTTP 请求信息流中再次返回到 HttpModule 中。HttpModule 是 ASP.NET 中包含的一组可由应用程序使用的模块。

当一个 HTTP 请求经过 HttpModule 容器传递到 HttpHandler 容器中时, ASP.NET Framework 会调用 HttpHandler 的相关方法来对这个 HTTP 请求进行真正的处理, 即对 ASP.NET 页面进行处理解析。处理完成后将结果继续经由 HttpModule 传递下去, 直至到达客户端。通过使用 HttpHandler 可以有效地降低服务器端资源的消耗, 减轻服务器的负担, 从而加快服务器端的响应速度。

□ Call Back

Call Back 是 ASP.NET 2.0 新增的开发方式。它要求页面实现 ICallbackEventHandler 接口。这样, 客户端页面中的脚本就可以采用 CallBack 机制调用服务器端方法, 从而避免开发人员直接操纵 XMLHTTP 对象, 在一定程度上简化了 Ajax 的开发。它是微软公司最早的 Ajax 策略, 目的是让用户像使用PostBack一样来使用它, 以达到异步更新数据的目的。但由于相对其他 Ajax 框架很难扩展, 所以应用受到了限制。

ICallbackEventHandler 接口用于指示该控件可以作为服务器的回调事件的目标。该接口在 .NET Framework 2.0 版中是新增的。它包括两个方法。RaiseCallbackEvent 方法执行对异步请求的服务器端处理, GetCallbackResult 方法返回异步请求的处理结果。

□ Ajax 框架

Ajax 框架现在已有很多。通常情况下, 将基于浏览器的应用框架称为客户端框架, 如 Prototype、DOJO、qooxdoo、Bindows 等。将基于服务器端的框架称为服务器端框架, 如 ASP.NET Ajax、Ajax.NET、WebORB for .NET、ComfortASP.NET、Ajax Aspects 等。通过使用 Ajax 框架进行 Web 开发可以提高效率, 并且代码稳定性好, 但同时也易于受框架的束缚。其中 ASP.NET