

# [Ajax]

## Web 2.0 快速入门与项目实践 (.net)

施伟伟 张 蓓 编著



人民邮电出版社  
POSTS & TELECOM PRESS



# Web 2.0

## 快速入门与项目实践 (.net)

施伟伟 张 蓓 编著

 人民邮电出版社  
POSTS & TELECOM PRESS

## 图书在版编目 (CIP) 数据

征服 Ajax: Web 2.0 快速入门与项目实践 (.net) 施伟伟, 张蓓编著.

—北京: 人民邮电出版社, 2006.4

ISBN 7-115-14706-X

I. 征... II. ①施...②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 (2006) 第 033663 号

## 内 容 提 要

利用 Ajax 实现的 Web 应用可以在不刷新页面的情况下与服务器进行异步的通信, 这个改变对于 Web 应用可以说是革命性的, 它使得开发人员得以摆脱原有单调的开发模式, 从而开发出更易用、交互性更强的 Web 应用。

本书共 11 章, 前两章介绍了 Ajax 技术的基本概念和相关的基础技术。第 3 章~第 5 章分别实现了基于 Ajax 的数据验证、自动完成功能以及一个基于 Ajax 的动态树形菜单, 这些例子虽然短小, 却代表了几种 Ajax 常见的应用场景。第 6 章~第 11 章介绍了一些完整的应用, 包括个性化主页、论坛主题搜索、无刷新聊天室、地图查看系统、相册浏览和 Web 版的 MSN。这些应用涵盖了 Ajax 开发的 7 项基本技术: XMLHttpRequest、DOM、XML、JavaScript、XHTML、CSS 和 XSLT。

本书的案例经过编者的精心设计, 尽可能减少与 Ajax 无关的内容, 让读者以最快的速度掌握 Ajax 开发技术。

本书适合具有一定 Web 开发经验的读者阅读。

### 征服 Ajax—Web 2.0 快速入门与项目实践 (.net)

◆ 编 著 施伟伟 张 蓓

责任编辑 张立科

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京顺义振华印刷厂印刷

新华书店总店北京发行所经销

◆ 开本: 800×1000 1/16

印张: 18.5

字数: 406 千字

2006 年 4 月第 1 版

印数: 1—6 000 册

2006 年 4 月北京第 1 次印刷

ISBN 7-115-14706-X/TP · 5370

定价: 36.00 元 (附光盘)

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223



## PREFACE

# 前 言

本书面向的读者是具有一定经验的 Web 开发人员。以前我们一直认为,在 Web 应用中客户端与服务器的交互只能通过表单的提交、服务器的处理后返回新的页面,最后客户端进行完全的页面刷新。这样的局面已经延续了相当长的时间,甚至连我们的客户也已经习惯于这种使用模式。但是我们不得不承认,Web 应用与桌面应用程序相比,缺少了很多交互的特性,浏览器在 Web 应用中所处的角色仅仅是一个呈现 HTML 代码的容器,而不是一个独立应用程序的宿主。

当 Ajax 这个名词出现在我们的视线中时,相信所有的 Web 开发人员都会感觉眼前一亮,这不就是我们一直在寻觅的技术吗?利用 Ajax 实现的 Web 应用可以在不刷新页面的情况下与服务器进行异步的通信。这一改变,对于 Web 应用来说是革命性的,它使得开发人员得以摆脱原有单调的开发模式,从而开发出更易用、交互性更强的 Web 应用。实际上,正是一些优秀的 Web 应用让我们认识了 Ajax:Google Maps、Gmail 无论在视觉效果还是易用性方面都对用户产生了巨大的冲击力,其中 Ajax 功不可没。这就是为什么 Ajax 会在短短的几个月内成为 Web 开发领域中炙手可热的名词。

应用 Ajax 开发 Web 应用对开发人员的技术,特别是 JavaScript 的编程技术,提出了更高的要求。目前介绍 Ajax 的书籍中,多是基于 XMLHttpRequest 的直接应用,并不针对特定的服务器端技术。尽管 Ajax 技术从本质上是属于客户端的技术,但是 Ajax 应用最精彩之处在于客户端与服务器的异步交互。在目前流行的各种服务器端技术中,ASP.NET 由于其易于快速上手、开发效率高等特点,已经成为开发中小型 Web 应用的首选。因此,本书的着眼点在于 Ajax 与 ASP.NET 这两种技术的结合,帮助读者在 ASP.NET 平台下快速构建基于 Ajax 的应用。

本书的前两章介绍了 Ajax 技术的基本概念及相关的基础技术,如果您不了解 Ajax,不知道 XMLHttpRequest 对象的使用方法,可以通过阅读这两章对 Ajax 技术进行初步的了解。

第 3 章~第 5 章实现了基于 Ajax 的数据验证、自动完成功能以及一个基于 Ajax 的

动态树形菜单。这些例子虽然短小，却代表了几种 Ajax 常见的应用场景，而且这几章的结果可以方便地移植到其他系统中。读者可以通过这几章的学习，了解如何在 ASP.NET 平台下开发基本的 Ajax 应用。

第 6 章~第 11 章的每一章介绍了一个完整的应用。它们分别是个性化主页、论坛主题搜索、无刷新聊天室、地图查看系统、相册浏览和 Web 版的 MSN。这些例子经过精心设计，尽可能减少与 Ajax 无关的内容，让读者以最快的速度掌握 Ajax 开发技术。这些示例涵盖了 Ajax 开发的 7 项基本技术：XMLHttpRequest、DOM、XML、JavaScript、XHTML、CSS 和 XSLT。通过这些章节的学习，读者可以全面地掌握 Ajax 开发技术。

本书中实例的一大特点是借助了第三方的框架或 JavaScript 脚本库来实现一些通用的功能。ASP.NET 服务器端 Ajax 框架包括 AjaxPro.NET、Magicajax 等当前流行的开发框架；在客户端的开发中也使用了一些开源的脚本库，合理地使用这些资源可以有效地提高开发效率，保证代码质量。

本书主要由施伟伟、张蓓编写，由于时间仓促和水平有限，书中错误和不妥之处在所难免，敬请读者批评指正。

编 者

2006 年 4 月



# CONTENTS

## 目 录

|                                                 |    |
|-------------------------------------------------|----|
| 第 1 章 认识 Ajax .....                             | 1  |
| 1.1 什么是 Ajax .....                              | 1  |
| 1.2 为什么使用 Ajax .....                            | 1  |
| 1.3 谁在使用 Ajax .....                             | 3  |
| 1.4 Ajax 应用场景 .....                             | 4  |
| 1.4.1 场景 1: 数据验证 .....                          | 4  |
| 1.4.2 场景 2: 按需取数据 .....                         | 4  |
| 1.4.3 场景 3: 自动更新页面 .....                        | 5  |
| 1.5 Ajax 应用实例 .....                             | 5  |
| 1.5.1 实例 1: Google Suggest .....                | 5  |
| 1.5.2 实例 2: Google Maps .....                   | 6  |
| 1.5.3 实例 3: A9.com .....                        | 7  |
| 1.5.4 实例 4: Gmail .....                         | 8  |
| 1.5.5 实例 5: Amazon 钻石搜索 .....                   | 10 |
| 1.6 Ajax 的竞争对手 .....                            | 10 |
| 1.7 小结 .....                                    | 12 |
| 第 2 章 Ajax 技术基础 .....                           | 13 |
| 2.1 Ajax 相关技术简介 .....                           | 13 |
| 2.1.1 XMLHttpRequest 对象 .....                   | 13 |
| 2.1.2 JavaScript .....                          | 15 |
| 2.1.3 DOM .....                                 | 16 |
| 2.1.4 XML .....                                 | 19 |
| 2.1.5 其他 .....                                  | 19 |
| 2.2 简单 Ajax 框架 .....                            | 19 |
| 2.3 Ajax 开发框架介绍 .....                           | 25 |
| 2.3.1 基于浏览器的应用框架 (Application Frameworks) ..... | 25 |

|       |                                        |     |
|-------|----------------------------------------|-----|
| 2.3.2 | 基于浏览器的应用框架 (Infrastructure Frameworks) | 28  |
| 2.3.3 | 基于服务器端的应用框架 (Java)                     | 30  |
| 2.3.4 | 基于服务器端的应用框架 (.NET)                     | 32  |
| 2.3.5 | 基于服务器端的应用框架 (其他)                       | 35  |
| 2.4   | 小结                                     | 38  |
| 第 3 章 | 基于 Ajax 的数据验证                          | 39  |
| 3.1   | 传统的数据验证方式                              | 39  |
| 3.2   | 基于 Ajax 的数据验证方式                        | 43  |
| 3.2.1 | 简单 Ajax 开发框架                           | 44  |
| 3.2.2 | 数据验证的实现                                | 51  |
| 3.3   | 应用 AjaxPro.NET 实现数据验证                  | 57  |
| 3.4   | 小结                                     | 62  |
| 第 4 章 | 基于 Ajax 的自动完成功能                        | 63  |
| 4.1   | 应用背景                                   | 63  |
| 4.2   | 自动完成功能的实现                              | 64  |
| 4.2.1 | 服务器端方法                                 | 64  |
| 4.2.2 | 客户端处理                                  | 65  |
| 4.3   | 自动完成文本框控件                              | 83  |
| 4.4   | 小结                                     | 87  |
| 第 5 章 | 基于 Ajax 的动态树形菜单                        | 89  |
| 5.1   | CSS 实现静态的树形菜单                          | 89  |
| 5.2   | 树形菜单的动态加载                              | 93  |
| 5.2.1 | 数据库设计                                  | 94  |
| 5.2.2 | 服务器端方法                                 | 94  |
| 5.2.3 | 客户端页面处理                                | 97  |
| 5.3   | 树形菜单的应用                                | 104 |
| 5.3.1 | 数据库设计                                  | 104 |
| 5.3.2 | 服务器端方法                                 | 105 |
| 5.3.3 | 客户端页面处理                                | 106 |
| 5.4   | 小结                                     | 115 |
| 第 6 章 | 基于 Ajax 的个性化主页                         | 117 |
| 6.1   | 数据库设计和 Web Service 的实现                 | 117 |
| 6.1.1 | 数据库设计                                  | 117 |

|       |                               |     |
|-------|-------------------------------|-----|
| 6.1.2 | Web Service 实现 .....          | 121 |
| 6.2   | 自动更新的栏目面板 .....               | 132 |
| 6.2.1 | Ajax 调用 Web Service .....     | 132 |
| 6.2.2 | 服务器端方法 .....                  | 133 |
| 6.2.3 | 客户端展现结果 .....                 | 134 |
| 6.3   | 可拖放的栏目面板 .....                | 137 |
| 6.3.1 | 拖放功能的实现 .....                 | 138 |
| 6.3.2 | 记录面板位置 .....                  | 138 |
| 6.4   | 使用褪色技术 .....                  | 142 |
| 6.5   | 小结 .....                      | 150 |
| 第 7 章 | 基于 Ajax 的论坛主题搜索 .....         | 151 |
| 7.1   | 数据库设计 .....                   | 151 |
| 7.2   | 应用 Magicajax 实现论坛主题搜索 .....   | 152 |
| 7.2.1 | Magicajax 的配置 .....           | 153 |
| 7.2.2 | 论坛搜索功能的实现 .....               | 154 |
| 7.3   | 应用 AjaxPro.NET 实现论坛主题搜索 ..... | 159 |
| 7.3.1 | XSLT 应用示例 .....               | 160 |
| 7.3.2 | 服务器端方法 .....                  | 166 |
| 7.3.3 | 客户端处理 .....                   | 168 |
| 7.4   | 论坛主题搜索功能的优化 .....             | 171 |
| 7.5   | 小结 .....                      | 179 |
| 第 8 章 | 基于 Ajax 的无刷新聊天室 .....         | 181 |
| 8.1   | 数据库设计 .....                   | 181 |
| 8.1.1 | 数据表设计 .....                   | 181 |
| 8.1.2 | 存储过程设计 .....                  | 183 |
| 8.2   | 聊天室系统实现 .....                 | 186 |
| 8.2.1 | 用户登录页面 .....                  | 187 |
| 8.2.2 | 聊天室主页面 .....                  | 190 |
| 8.3   | 小结 .....                      | 207 |
| 第 9 章 | 基于 Ajax 的地图查看 .....           | 209 |
| 9.1   | 地图查看系统功能介绍 .....              | 209 |
| 9.2   | 鹰眼效果的实现 .....                 | 210 |
| 9.3   | 地图显示区域的动态更新 .....             | 212 |
| 9.3.1 | 直接输出 .....                    | 212 |



|                                  |     |
|----------------------------------|-----|
| 9.3.2 栅格化图片输出 .....              | 215 |
| 9.4 地图详细信息的查看 .....              | 226 |
| 9.5 地图查看的优化 .....                | 229 |
| 9.6 小结 .....                     | 233 |
| 第 10 章 基于 Ajax 的相册浏览 .....       | 235 |
| 10.1 相册浏览系统功能分析 .....            | 235 |
| 10.2 数据库设计 .....                 | 236 |
| 10.3 分类图片信息的读取 .....             | 237 |
| 10.4 相册浏览的实现 .....               | 240 |
| 10.4.1 图片的切换显示 .....             | 240 |
| 10.4.2 图片信息的切换 .....             | 244 |
| 10.5 小结 .....                    | 253 |
| 第 11 章 基于 Ajax 的 Web 版 MSN ..... | 255 |
| 11.1 Web 版 MSN 功能分析 .....        | 255 |
| 11.2 数据库设计 .....                 | 255 |
| 11.2.1 数据表设计 .....               | 255 |
| 11.2.2 存储过程设计 .....              | 257 |
| 11.3 Web 版 MSN 的实现 .....         | 261 |
| 11.3.1 登录页面 .....                | 262 |
| 11.3.2 主页面 .....                 | 266 |
| 11.3.3 消息发送页面 .....              | 278 |
| 11.4 小结 .....                    | 288 |

## 第 1 章 认识 Ajax

在介绍 Ajax 技术之前,首先谈一谈 Web 应用与桌面应用。Web 应用的优点在于部署和维护过程非常简单,而桌面应用程序则具有丰富的界面和快速的响应能力。一直以来人们都是根据实际需要,在两种应用之间进行选择,通常只能获得一种应用的优势,但是今天情况却逐渐发生着变化:请看看“Google Suggest”,它根据用户的输入实时显示着建议的条目;再看看“Google Maps”,用鼠标挪动、放大或者缩小地图,响应速度也非常快,无需等待页面的刷新。类似的应用还有很多,例如 Gmail、Orkut 和 Amazon 的 A9.com 搜索,它们都是基于 Ajax 技术的 Web 应用。

### 1.1 什么是 Ajax

Ajax 是异步 JavaScript 和 XML (Asynchronous JavaScript and XML) 的英文缩写。“Ajax”这个名词的发明人是 Jesse James Garrett,而大力推广并且使 Ajax 技术炙手可热的是 Google。Google 发布的 Gmail、Google Suggest 等应用最终让人们了解了什么是 Ajax。

Ajax 的核心理念在于使用 XMLHttpRequest 对象发送异步请求,最初为 XMLHttpRequest 对象提供浏览器支持的是微软公司。早在 1998 年,微软公司开发 Web 版 Outlook 时,就已经以 ActiveX 控件的方式为 XMLHttpRequest 提供了支持(当然,仅限于 Microsoft Internet Explorer)。

实际上, Ajax 不是一种技术,而是几种技术。每种技术都具有独特之处,合在一起就形成了一个功能强大的新技术。Ajax 包括:

- XHTML 和 CSS;
- 使用文档对象模型 (Document Object Model) 进行动态显示和交互;
- 使用 XML 和 XSLT 进行数据交互和操作;
- 使用 XMLHttpRequest 进行异步数据接收;
- 使用 JavaScript 将它们绑定在一起。

### 1.2 为什么使用 Ajax

传统的 Web 应用采用同步交互过程,这种情况下,用户首先向 Web 服务器发送一个请

求, 然后 Web 服务器根据用户请求的内容, 执行相应的任务, 并向用户返回结果, 如图 1-1 所示。这是一种不连贯的用户体验, 在服务器处理请求时, 用户只能等待, 此时浏览器显示的页面是空白的, 这也就是通常所说的“白屏”现象。

自从采用 HTML 进行 Web 传输和呈现以来, 无论是基于哪种服务器技术 (ASP、ASP.NET、JSP 或 PHP), Web 应用中采用的都是如图 1-1 所示的处理模式。当页面内容比较少, 或者服务器处理时间较短的时候, 采用这种模式似乎没有什么不妥。但是, 如果页面内容很多, 服务器的响应时间较长, 对于用户来说就难以接受了。根据调查, 一个网页加载的时间如果超过 4~5 秒, 那么大多数用户将不会等待, 可能会选择切换到其他窗口或者直接关闭该页面。

传统Web应用程序模型（同步）

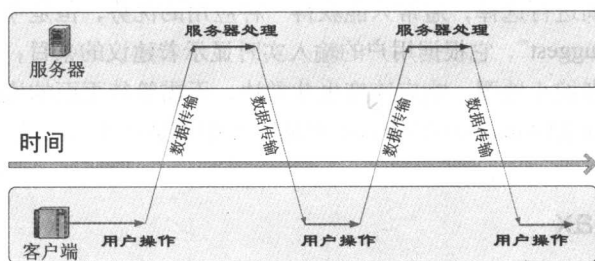


图 1-1 传统 Web 应用程序模型

此外, 用户在某些时候仅仅需要改变页面中某部分的数据, 但是他不得不刷新整个页面。尤其在人机交互较为频繁的应用系统中, 这种现象屡见不鲜, 这显然是和人性化的软件设计原则相违背的。如何减少用户的等待时间, 提高系统的可用性呢? 现在除了优化代码、调整服务器配置之外, 还可以选择使用 Ajax 技术。

与传统的 Web 应用不同, Ajax 采用了异步交互的方式。它在用户和服务器之间引入了一个中间媒介, 从而改变了同步交互过程中的“处理—等待—处理—等待”模式。用户的浏览器在执行任务时即装载了 Ajax 引擎。该引擎是用 JavaScript 语言编写的, 通常位于一个隐藏的框架中, 负责转发用户界面和服务器之间的交互。Ajax 引擎允许用户和应用系统之间的交互以异步的方式进行, 独立于用户与 Web 服务器之间的交互。现在, 可以用 JavaScript 调用 Ajax 引擎来产生一个 HTTP 的用户请求, 数据编辑、页面导航和数据验证等操作不再需要重新加载整个页面, 可以通过 Ajax 引擎向 Web 服务器发送请求。基于 Ajax 的 Web 应用程序的模型如图 1-2 所示。

使用 Ajax 可以带来的好处有以下几方面。

- 减轻服务器的负担。Ajax 的原则是“按需取数据”, 可以最大程度地减少冗余请求, 减轻服务器的负担。

AJAX Web应用程序模型（非同步）

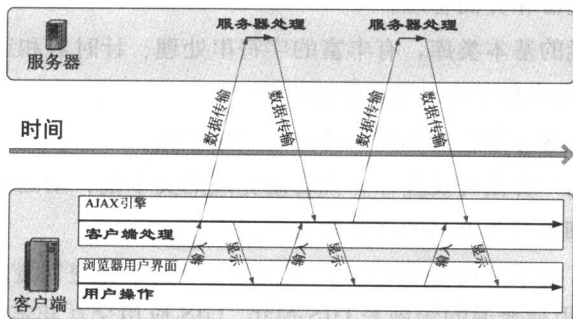


图 1-2 基于 Ajax 的 Web 应用程序模型

- 无需刷新页面，减少用户心理和实际的等待时间。特别是在读取大量数据时，不会像刷新页面那样出现白屏的情况，Ajax 使用 XMLHttpRequest 对象发送请求并且得到服务器响应，在不重新载入整个页面的情况下，用 JavaScript 操作 DOM 更新页面。因此在读取数据的过程中，用户所面对的不是白屏，是原来的页面内容（也可以加一个“Loading”的提示框让用户知道目前正在读取数据），只有在数据接收完毕之后才更新相应部分的内容。这种更新是瞬间的，用户几乎感觉不到。
- 带来更好的用户体验。
- 可以把以前一些服务器负担的工作转嫁到客户端，利用客户端闲置的能力来处理，减轻服务器负担，充分利用带宽资源，节约空间和宽带租用成本。
- 可以调用外部数据。
- 基于标准化的并被广泛支持的技术，不需要下载插件或者小程序。
- 进一步促进页面呈现与数据的分离。

### 1.3 谁在使用 Ajax

在 Ajax 的应用和推广方面，Google 当仁不让是表率，前面提到很多基于 Ajax 的应用都是由 Google 推出的。

微软也在积极开发更为完善的 Ajax 应用：它即将推出代号为 Atlas 的 Ajax 开发包。Atlas 的功能将超越 Ajax 本身，包括整合 Visual Studio 的调试功能（客户端 JavaScript 脚本的调试一直是困扰开发人员的难题）。另外，新的 ASP.NET 控件将使客户端控件和服务端代码的捆绑更为简便。Atlas 客户端脚本框架（Atlas Client Script Framework）也使网页及相关项目的交互更为便利。

微软近期宣布 Atlas 客户端脚本框架将包含以下内容（详细内容可以查看微软网站）：

- 一个可扩展的核心框架，它添加了 JavaScript 功能，如生命同时期管理、继承管理、多点传送处理器和界面管理；
- 一个常见功能的基本类库，有丰富的字符串处理、计时器和运行任务的功能；
- 为 HTML 附加动态行为的用户界面框架；
- 一组用来简化服务器连通和网络访问的网络堆栈；
- 一组丰富的用户界面开发控件，如自动完成的文本框、动画和拖放；
- 处理浏览器脚本行为差异的浏览器兼容层面。

微软已经将 Ajax 技术应用在 MSN Space 网站上，其中提交回复的功能就是利用 Ajax 实现的。目前，Ajax 应用最普遍的领域是 GIS-Map，GIS 应用交互非常频繁，它强调快速响应，Ajax 的特点正好可以满足这类需求。

## 1.4 Ajax 应用场景

Ajax 不是万能的，在适宜的场合使用 Ajax，才能充分发挥它的长处，改善系统性能和用户体验，绝不可以为了技术而滥用。Ajax 的特点在于异步交互、动态更新 Web 页面，因此它的适用范围是交互较多、频繁读取数据的 Web 应用。现在来看几个 Ajax 的应用实例，读者可以了解如何使用 Ajax 技术改进现有的 Web 应用系统。

### 1.4.1 场景 1：数据验证

在填写表单内容时，需要保证数据的惟一性（例如新用户注册填写的用户名），因此必须对用户输入的内容进行数据验证。数据验证通常有两种方式：一种是直接填写，然后提交表单，这种方式需要将整个页面提交到服务器端进行验证，整个过程不仅时间长而且造成了服务器不必要的负担；第二种方式是改进了的验证过程，用户可以通过点击相应的验证按钮，打开新窗口查看验证结果，但是这样需要新开一个浏览器窗口或者对话框，还需要专门编写验证的页面，比较耗费系统资源。而使用 Ajax 技术，可以由 XMLHttpRequest 对象发出验证请求，根据返回的 HTTP 响应判断验证是否成功，整个过程不需要弹出新窗口，也不需要将整个页面提交到服务器，快速而又不加重服务器负担。

### 1.4.2 场景 2：按需取数据

分类树或树形结构在 Web 应用系统中使用得非常广泛，例如部门结构、文档的分类结构常常使用树形控件呈现。以前每次对分类树的操作都会引起页面重载，为了避免这种情况出现，一般不采用每次调用后台的方式，而是一次性将分类结构中的数据全部读取出来并写入数组，然后根据用户的操作，用 JavaScript 来控制节点的呈现，这样虽然解决了操作响应速度、不重载页面以及避免向服务器频繁发送请求的问题，但是如果用户不对分类树进行操作

或者只对分类树中的一部分数据进行操作的话（这种情况是很普遍的），那么读取的数据中就会有相当大的冗余，浪费了用户的资源。特别是在分类结构复杂、数据量庞大的情况下，这种弊端就更加明显了。

现在应用 Ajax 改进分类树的实现机制。在初始化页面时，只获取第一级子分类的数据并且显示；当用户点开一级分类的某一节点时，页面会通过 Ajax 向服务器请求当前分类所属的二级子分类的所有数据；如果再继续请求已经呈现的二级分类的某一节点时，再次向服务器请求当前分类所属的三级子分类的所有数据，以此类推。页面会根据用户的操作向服务器请求它所需要的数据，这样就不会存在数据的冗余，减少了数据下载总量。同时，更新页面时不需要重载所有内容，只更新需要更新的那部分内容即可，相对于以前后台处理并且重载的方式，大大缩短了用户的等待时间。

### 1.4.3 场景 3：自动更新页面

在 Web 应用中有很多数据的变化是十分迅速的，例如最新的热点新闻、天气预报以及聊天室的聊天内容等。在 Ajax 出现之前，用户为了及时了解相关的内容必须不断刷新页面，查看是否有新的内容变化，或者页面本身实现定时刷新的功能（大多数聊天室页面就是这样做的）。有可能会发生这种情况：有一段时间网页的内容没有发生任何变化，但是用户并不知道，仍然不断地刷新页面；或者用户失去了耐心，放弃了刷新页面，却很有可能在此时有新的消息出现，这样就错过了第一时间得知消息的机会。

应用 Ajax 可以改善这种情况，页面加载以后，会通过 Ajax 引擎在后台进行定时的轮询，向服务器发送请求，查看是否有最新的消息。如果有则将新的数据（而不是所有数据）下载并且在页面上进行动态的更新，通过一定的方式通知用户（实现这样的功能正是 JavaScript 的强项）。这样既避免了用户不断手工刷新页面的不便，也不会因为重复刷新页面造成资源浪费。

## 1.5 Ajax 应用实例

尽管 Ajax 的概念提出的时间还不长，但是目前已经出现了许多基于 Ajax 的应用。这一节中将介绍其中的几个，读者也可以访问相应的网站，亲身体会 Ajax 应用具有的独特的用户体验。

### 1.5.1 实例 1：Google Suggest

Google Suggest（如图 1-3 所示）实在是一个值得称道的 Ajax 应用，它不仅很好地设置了下拉区，还会在输入框中自动输入最有可能的答案，并且将非用户输入的部分置灰，在下拉区中可以使用上下箭头或者鼠标选择搜索词。下拉区为每个可能的搜索词都提供了一些结果，用户可以清楚地了解当前输入的搜索词是否合适。

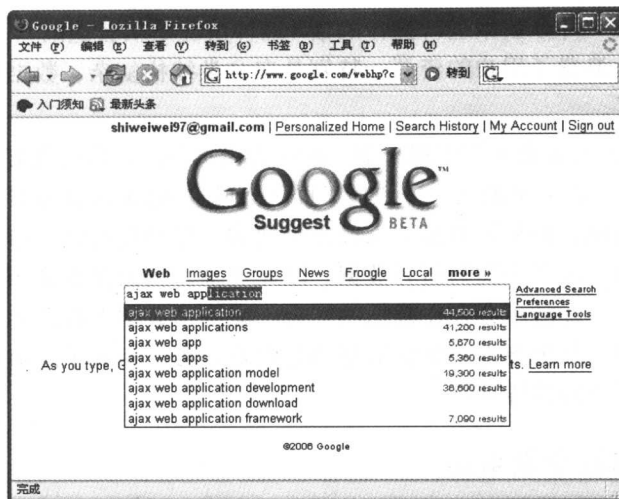
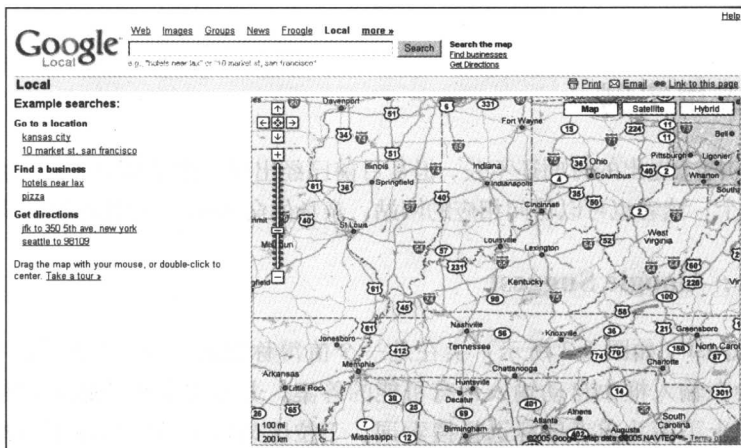


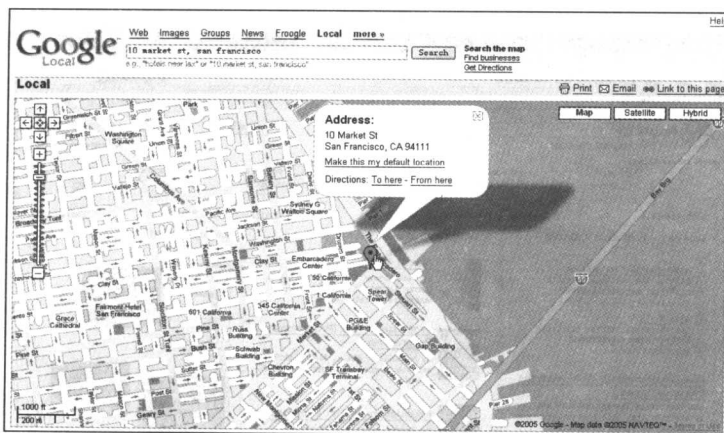
图 1-3 Google Suggest

## 1.5.2 实例 2: Google Maps

Google Maps (如图 1-4(a)所示) 是一个地图搜索的 Web 应用。在地图展示区中, 用户可以用鼠标拖动地图, 或者用滚轮放大、缩小地图。在这个操作过程中用户可以看到, 原来没有展示的区域会暂时处于空白状态, 不久从服务器端传回的数据就可以将空白部分填满。但是整个下载数据的过程并不影响用户, 用户可以继续其他的操作。Google Maps 中还有一个很有趣的功能就是动态的信息提示, 如图 1-4(b)所示。当鼠标移到相应位置上时, 会出现有关该位置的详细信息。注意, 这些信息是即时获取的, 而不是事先下载到本地的, 实际上这也是 Ajax 按需取数据原则的一个体现。



(a)

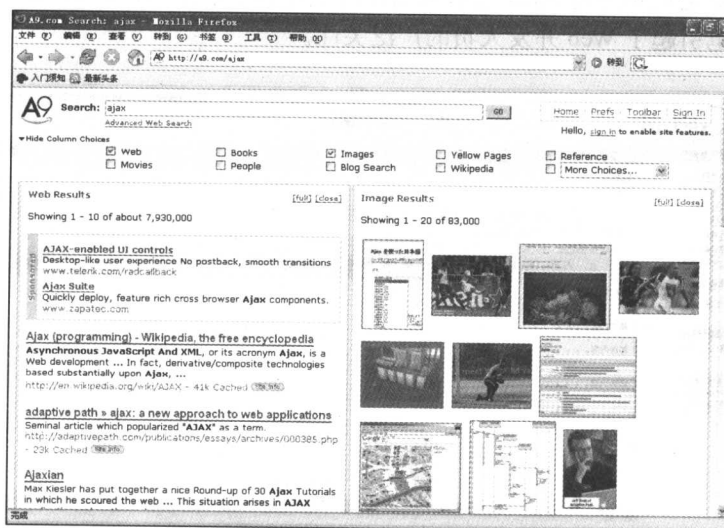


(b)

图 1-4 Google Maps

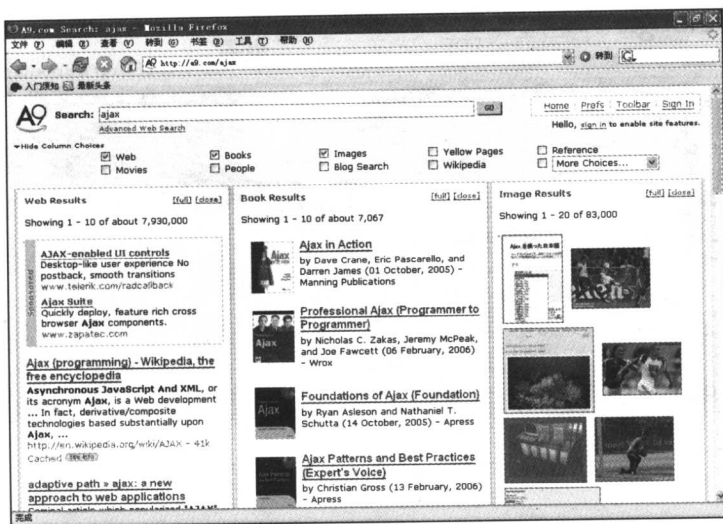
### 1.5.3 实例 3: A9.com

A9.com (见图 1-5) 是 Amazon 的搜索引擎, 其特点是搜索选项可以随时增加、去除。如图 1-5(a)所示是仅搜索“Web 页”和“图像”的结果, 如果这时用户还想看一看当前的搜索词在“图书”中有哪些结果, 只需勾上“books”选项, 页面会自动变为图 1-5(b)所示, 及时加载“图书”相关的搜索结果。由于应用了 Ajax 技术, 原有的搜索结果仍然在网页上保留, 只是添加了在“图书”范围内搜索并返回的结果, 同时页面的布局发生了相应改变。当然所有这些过程中, 页面只是局部更新, 而不是刷新整个页面。



(a)



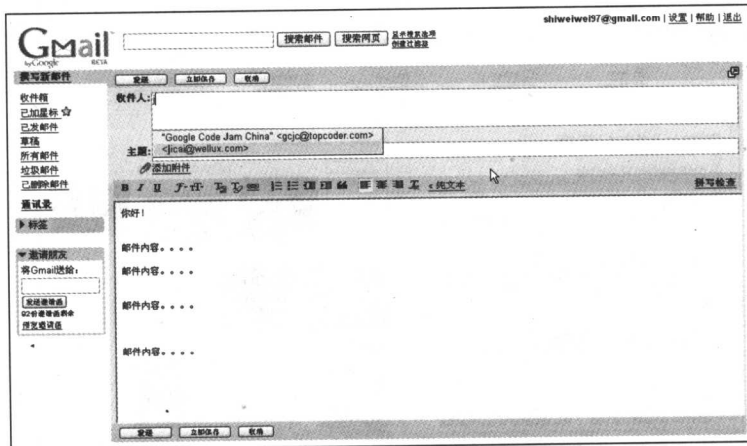


(b)

图 1-5 A9.com

## 1.5.4 实例 4: Gmail

Gmail (如图 1-6 所示) 是 Google 推出的电子邮件系统, 从发布之日起 Gmail 就受到了业界极大的关注。原因之一是其超大的容量 (2.5GB, 并且在不断增长中), 以及不同于其他电子邮件系统的邀请注册方式, 这是 Google 推广策略的独到之处, 和技术本身关系并不大。另一个重要原因则是该系统将 Ajax 技术应用得淋漓尽致, 给用户使用 Gmail 邮件系统带来了很多便利, 同时也引起了 Web 开发人员的广泛关注。



(a)