



Beginning Ajax

Ajax 入门经典



(美) Chris Ullman
Lucinda Dykes

徐 璐

著

译



清华大学出版社

TP393.09/202

2008

Ajax 入门经典

(美) Chris Ullman 著
Lucinda Dykes 译
徐璐

清华大学出版社

北 京

Chris Ullman, Lucinda Dykes

Beginning Ajax

EISBN: 978-0-470-10675-4

Copyright © 2007 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2007-2837

本书封面贴有 Wiley 公司防伪标签, 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 入门经典/(美)乌尔曼(Ullman, C.), (美)戴科斯(Dykes, L.) 著; 徐璐 译.

—北京: 清华大学出版社, 2008.8

书名原文: Beginning Ajax

ISBN 978-7-302-18003-6

I. A… II. ①乌…②戴…③徐 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(2008)第 095571 号

责任编辑: 王 军 王 婷

装帧设计: 孔祥丰

责任校对: 成凤进

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: 010-62776969, 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: 三河市溧源装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 29.25 字 数: 712 千字

版 次: 2008 年 8 月第 1 版 印 次: 2008 年 8 月第 1 次印刷

印 数: 1~4000

定 价: 58.00 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: 025904-01

作者简介

Chris Ullman 是一名自由 Web 开发人员和技术作家，有着多年的 ASP/ASP.NET 工作经验。具有计算机科学背景的他最初是 UNIX/Linux 方面的权威，在 1997 年 ASP 盛行时被 Microsoft 技术所吸引。他从 Wrox 出版社的 ASP 指南开始介入图书编撰，此后主编或参与创作了多达 25 本书籍，最值得一提的就是他是 Wrox 畅销书 Beginning ASP/ASP.NET 1.x 系列的主要作者。他也参与编写了关于 PHP、ColdFusion、JavaScript、Web 服务、C#、XML 和其他因特网技术书籍的一些章节。2001 年 8 月，他离开全职工作的 Wrox，开始涉足于 VB.NET/C# 编程和 ASP.NET 开发，并于 2003 年 4 月起经营自己的 CUASP 咨询有限公司。他还负责维护很多网站，从 <http://www.cuasp.co.co.uk> (他自己的“工作”网站) 到 <http://www.atomicwise.com> (包含他在音乐和艺术方面创作的精选)。目前他把时间用于陪伴家人和为他的音乐项目 Open E 编写电子配乐。

Lucinda Dykes 是一名自由 Web 开发人员、教师兼技术作家，从 1994 年就开始编写代码和开发 Web 站点。她最初的职业开始于医学高科技领域，但她后来离开医学而转向发展自己在技术和 Web 方面的兴趣。她在公司 Zero G Web 多年，致力于用 JavaScript 进行客户端开发，并在 eclasses.org 在线为来自全球各地的学生讲授 JavaScript 课程。她也撰写了大量关于 XML、XHTML 和用 Dreamweaver 进行 Web 应用程序开发的书籍。

前言

Ajax 在过去的两年成为了一个非常时髦的词汇。它经常和另一个时髦词——Web 2.0 被同时提及。这两者都不是什么具体的或者可下载的事物，但它们的出现反映了 Web 的演化。Web 2.0 是一种新发展，它更是一种态度和概念。一些新事物的出现共同促成了这种改变的浪潮，比如 wiki、博客、新闻反馈、第三方应用程序编程接口(APIs)以及 Web 服务。Ajax 则是主要的促成者之一。

Ajax 本身并不是一种技术，而是一个术语，用来描述如何结合使用几种现存的技术，比如 JavaScript、文档对象模型(DOM)、可扩展标记语言(XML)，来创建交互性更强的 Web 应用程序，并且实现当页面的一部分改变时不需要刷新整个页面的功能。

虽然术语 Ajax 一词是最近才出现，实际上 Ajax 应用程序中使用的技术已经存在了好些年。有些编程人员已经使用 Ajax 风格的技术或技巧至少 5 年了。在过去一年中使用 Ajax 风格的技术开发的 Web 站点数量大幅增多。此外许多新的工作职位要求编程人员知道如何编写 Ajax 风格的应用程序。本书就是帮助开发人员理解术语 Ajax 背后的核心技术，以使人们能够开始建立使用 Ajax 技术的站点。

尽管很多人都听说过 Ajax，但很少有人能够利用这些技术编写应用程序。Ajax 模糊了以前传统意义上的前端开发人员和服务器端开发人员之间的界限，它迫使人们改进创建应用程序的方式以及用户接口提供的功能。

Ajax 不要求新的软件、新的服务器或新的工具。它合理利用已有的环境，对服务器上的一切行为做出挑战性的理解，为原有的概念赋予新的功能，从根本上改变了 Web 工作的方式。

本书读者对象

本书讲述如何根据 Ajax 原则来创建应用程序。无论 Internet Explorer(IE)、Firefox 或者 Safari 到底哪一个是更强的浏览器，本书的例子可以在所有主流浏览器上工作。由于其固有的复杂性，Ajax 对于初学者来说并不容易上手。所以希望读者熟悉以下客户端技术：

- HTML 或 XHTML
- JavaScript
- 层叠样式表(CSS)

读者需要至少熟悉以下两种服务器端技术之一：

- PHP
- ASP.NET

服务器端的例子会用 PHP 和 ASP.NET/C#两种语言给出,但不需要对这两种技术有深入了解。

我们并不要求读者了解以下所示技术,因为本书会对它们做完整的介绍(当然了解其中之一或一部分会有所帮助)。

- 文档对象模型(DOM)
- XML
- XPath
- 可扩展样式表转换语言(XSLT)
- Web 服务(REST 和 SOAP)

像其他 Wrox 的入门书籍那样,我们在一章中讨论的概念会在其他章节中引用或延伸。

本书主要内容

本书讨论 Ajax 是什么、它对 Web 开发人员的意义和 Ajax 应用程序背后的相关技术。前面的章节探讨 Ajax 技术的优点和缺点,快速回顾了 JavaScript 技术。通过学习本书,您将发现 Ajax 应用程序是如何跨越客户端和服务端开发技术的。您会研究一些通用的 Ajax 模式,并看到 Ajax 可以如何联系现有技术,如 XSLT、Web 服务和 DOM。最后的章节给出一个深入学习创建 Ajax 应用程序的示例。

以下分别列出每章要讲述的内容:

- 第 1 章“Ajax 简介”——本章客观地给出了 Ajax 是什么,它的优点是什么,它的缺点或潜在的缺点是什么的描述。本章还研究了不同的技术,并给出一个可靠而开放的例子,该示例从 XML 后端显示一个动态菜单。
- 第 2 章“重温 JavaScript”——在开始介绍创建 Ajax 应用程序的细节之前,所有的读者都应处于同样的起点。本章快速回顾了 JavaScript 技术并讨论了 DOM,及它在 Ajax 应用程序中举足轻重的原因。
- 第 3 章“Ajax 和服务端技术”——尽管 Ajax 其主要内容是关于用 JavaScript 构建应用程序和调用处于底层的 service,但有关服务端如何接收、处理和返回数据的概念也非常重要。本章介绍了 XMLHttpRequest 对象以及它是如何调用 ASP.NET、PHP 和 Java 应用程序的,也讨论了它把信息返回给客户机的不同方式和格式。
- 第 4 章“Ajax 技术”——本章深入揭示了 XMLHttpRequest 对象的细节。讨论了所有可用于 Ajax 的技术,而不管它们是用于 XMLHttpRequest 对象、隐藏帧、隐藏的内联帧、动态图像载入还是用于动态脚本载入。该讨论也涉及到这些技术可能被用到的一些场景。
- 第 5 章“XML 的使用”——XML 实际上已经成为一个用结构化方式保存和发送数据的标准,也是 Ajax 技术中使用 XMLHttpRequest 对象的准则之一。尽管该对象并不要求只使用 XML,实际上它基本上是使用 XML。本章还探讨了创建 XML 文档的规则以及如何对 XML 文档进行有效的浏览和查询。

- 第 6 章“调试与错误处理”——集成开发环境(IDE)比如 Visual Studio、.NET 和 Dreamweaver 给用户提供了用于调试的工具集和安全机制。然而没有它们也可以定位错误。本章介绍了如何最好地手动调试 JavaScript 应用程序以及如何编写错误处理代码,那样您的 Ajax 应用程序就可以处理任何不确定的结果。
- 第 7 章“Web 服务、API 和 Mashup”——Web 2.0 这个七巧板中的另一块是大公司或集团的应用功能的开放。任何开发人员都可以自由地通过 Web 服务或者 API 把地图、图片、天气预报或者音乐播放列表嵌入到应用程序中。本章将介绍如何把这些新特性结合到 Ajax 应用程序中来创建 mashup。
- 第 8 章“XSLT 和 XPath”——XSLT 和 XPath 是基于 XML 的技术,它们提供更有效和灵活的 XML 文档查询和输出方式(而不是用 JavaScript 和 DOM)。本章通过使用这两种语法来对它们做简明的介绍,并展示如何使用它们来创建一个丰富而又无缝的用户接口。
- 第 9 章“模式”——编程总是和代码的模块化以及重用相关。针对业务中重复出现的问题,用一些测试过的、可靠的技术创建的解决方案就是模式。Ajax 应用程序在特定的情况下借用了这些概念(比如页面预载入和表单验证)。本章介绍了一些 Ajax 可用来实现的一些通用模式。
- 第 10 章“使用外部数据”——本章讲述了如何使用外部输入,比如 RSS 和 Atom,以及如何创建应用程序去使用数据源,该数据源会以固定时间间隔提供数据。此外,还讨论了示例种子(feed)的结构以及一个应用程序如何读取这样的种子。
- 第 11 章“JSON”——本章介绍了 JavaScript 发送数据的另一种方式:JavaScript 对象表示法(JSON)。它返回数组和对象字面量的集合。这里讨论了这种格式可以用来解决什么问题以及如何构建一个组件来返回它。
- 第 12 章“高级示例:可排序列表”——最后一章给出一个较大的示例应用程序,它用到 Scriptaculous 和本书中学到的技术,包括 XMLHttpRequest 对象、DOM 和键盘事情处理程序。
- 附录 A“习题解答”——包含所有章节的习题答案。
- 附录 B“Ajax 资源:架构和库”——指导如何安装和使用一些最常用的架构和库,如 Prototype、Scriptaculous、Dojo 以及更多。
- 附录 C“JavaScript 资源”——列出一些针对 JavaScript 的有用的博客、教程、文章和网站。
- 附录 D“JavaScript 语言参考”——JavaScript 的对象和语言结构的概述。

本书的结构

本书通过可运行的示例和详细的解释循序渐进地讲述概念,演示如何创建 Ajax 应用程序。尽管并不是每一章都依赖于前面章节的知识,但大多数章节是前后关联的。所以理解这些概念的最佳学习途径是从前向后浏览本书。Wrox 的入门书籍都是以陈述的方式来给用户提供一个系统而又友好的学习途径。即通常先告诉您将要做什么,接着通过一个示例来指导

您, 然后解释您学到的内容。在本书的两个作者的密切合作(编辑的有力支持)下, 本书将给您提供一个风格一致而又完整的教程。

使用本书的要求

为了更好地使用这本书, 需要安装以下软件:

- 最简配置的 Windows XP Professional 或某个版本的 Linux
- 至少安装 IIS 或 Apache 这两个 Web 服务器之一
- 至少安装 PHP(版本 5)或 ASP.NET(版本 2.0)这两个服务器端技术之一
- Internet Explorer 版本 6 或 7 和 Mozilla Firefox 版本 2
- 文本编辑器(比如记事本或 emacs 编辑器)

尽管 Ajax 应用程序不一定必须要调用服务器端页面, 但如果不这样的话, 它们的可用性和潜在的功能就大打折扣。本书的很多例子都要求服务器运行 PHP 或者 ASP.NET。每个例子都给出两个版本, 除极个别的例子以外。当然这种情况下会预先给出不这么做的理由。

本书约定

为了帮助您最有效地利用本书, 下面介绍全书中使用的大量约定。

试一试

“试一试”部分是一些您应该动手做的练习, 它有以下特点:

- (1) 它们由一组步骤组成。
- (2) 每一步骤都有一个编号。

(3) 按照这些步骤编辑您的代码。可以自己输入代码或者从 www.wrox.com 网站下载。偶尔可能会有附加的内容在网站上有提供, 但是书中没有提及(比如图像)。这些内容对于让示例工作来说不是必需的。

示例的说明

在每个“试一试”部分之后, 会详细解释输入的代码。

另外, 代码以两种不同的方式出现:

在代码示例中, 新的或重要的代码用灰色的背景突出显示

对于那些不太重要的或者以前已经出现过的代码则不用灰色突出显示。

源代码

在完成本书的示例时, 可以选择手动输入代码或者使用本书附带的源代码文件。本书用到的所有源代码可以从 www.tupwk.com.cn/downpage 下载, 也可以从 www.wrox.com 下载。进入该站点后, 只需找到本书的名称(使用 Search 框或者书名列表), 单击本书的详细

页面上的 Download Code 链接, 就可以得到本书所有的源代码。

注意:

由于很多书有相似的名称, 所以用 ISBN 搜索更为容易。本书的 ISBN 是 978-0-470-10675-4。

下载了代码后, 用您喜欢的压缩工具把它解压缩。此外也可以去 Wrox 的主下载页面 www.worx.com/dynamic/books/download.aspx 找到本书或其他 Wrox 出版的书的代码。

勘误表

尽管我们竭尽所能来确保在正文和代码中没有错误, 但人无完人, 错误难免会发生。如果您在 Wrox 出版的书中发现了错误(比如拼写错误或者代码错误), 我们将非常感谢您的反馈。发送勘误表将节省其他读者的时间, 同时也会帮助我们提供更高质量的信息。

到 www.wrox.com 站点上, 用 Search 框或者标题列表找到本书的名称, 在详细页面上点击 Book Errata 链接就能找到本书的勘误表。在这个页面中可以看到所有被提交的本书的勘误表, 它们是由 Wrox 的编辑发布的。在 www.worx.com/misc-pages/booklist.shtml 中有完整的书的列表, 其中包括每本书的勘误表。

如果您在书的勘误表页面上没有看到您发现的错误, 可以到 www.wrox.com/contact/techsupport.shtml 上填写表单, 把您发现的错误发给我们。我们会检查这些信息, 如果属实就把它添加到本书的勘误表页面上, 并在本书随后的版本中更正错误。

p2p.wrox.com

如果想和作者或者其他讨论, 请加入在 <http://p2p.worx.com> 的 P2P 论坛。该论坛是基于 Web 的系统, 您可以发布关于 Wrox 出版的书和相关技术的消息, 与其他读者或技术人员交流。该论坛有预定功能, 在您选择的感兴趣的主题有新帖子时, 会邮件通知。Wrox 的作者、编辑、其他业界专家和像您一样的读者都会出现在这些论坛中。

在 <http://p2p.wrox.com>, 您会找到很多不同的论坛, 它们不但有助于您阅读本书, 还有助于您开发自己的应用程序, 加入论坛的步骤为:

- (1) 到 <http://p2p.wrox.com> 上单击 Register 链接。
- (2) 阅读使用说明, 单击 Agree 按钮。
- (3) 填写加入必需的信息和其他您愿意提供的信息, 单击 Submit 按钮。
- (4) 您将收到一封 email, 描述如何验证您的账户和完成加入过程。

注意:

不加入 P2P 也可以阅读论坛里的消息。但是如果要发布自己的消息, 就必须加入。

加入之后，就可以发布新的消息和回复其他用户发布的消息。可以随时在 Web 上阅读论坛里的消息。如果想让某个论坛的新消息以 E-mail 的方式发给您，可以单击论坛列表里论坛名字旁边的 **Subscribe to this Forum** 图标。

要了解如何使用 Wrox P2P 的更多信息，请阅读 **P2P FAQs**，其中回答了论坛软件如何使用的问题，以及许多与 P2P 和 Wrox 出版的书相关的问题。要阅读 FAQs，单击任何 P2P 页面里的 **FAQ** 链接即可。

目 录

第 1 章	Ajax 简介	1
1.1	什么是 Ajax	2
1.1.1	Ajax 的应用	3
1.1.2	Ajax 的全称	8
1.2	Ajax 应用程序模型	15
1.2.1	使用 Ajax 的原因	17
1.2.2	Ajax 不适合的场合	18
1.2.3	使用 Ajax 的条件	19
1.3	创建自己的 Ajax	19
1.4	本章小结	29
1.5	练习	29
第 2 章	重温 JavaScript	31
2.1	核心 JavaScript	31
2.1.1	语法	31
2.1.2	变量	32
2.1.3	运算符	34
2.1.4	语句	36
2.1.5	函数	39
2.2	面向对象的 JavaScript	40
2.2.1	内置对象	40
2.2.2	浏览器对象	41
2.2.3	用户自定义对象	42
2.3	文档对象模型	46
2.3.1	文档的家谱树结构	47
2.3.2	文档的节点树结构	47
2.3.3	用来访问对象的 DOM 方法	48
2.3.4	创建节点	49
2.3.5	另一种方案: innerHTML	51
2.4	JavaScript 和事件	52
2.4.1	事件模型	52
2.4.2	事件注册程序	53
2.4.3	事件对象	55
2.5	本章小结	58

2.6	练习	59
第 3 章	Ajax 和服务器端技术	61
3.1	Ajax 和服务器端技术	61
3.1.1	表单和 HTML 控件	62
3.1.2	表单的提交模型	62
3.1.3	Ajax/JavaScript 提交模型	63
3.2	服务器端的情况	64
3.2.1	向服务器提交数据	64
3.2.2	服务器接收请求	65
3.3	编写 HTTP 响应	65
3.4	服务器端技术	70
3.4.1	ASP.NET	71
3.4.2	使用 AJAX 和 ASP.NET 的示例	72
3.5	PHP	82
3.6	Java Servlet	88
3.7	应当使用哪种技术	93
3.8	本章小结	94
3.9	练习	94
第 4 章	Ajax 技术	95
4.1	XMLHttpRequest 对象	95
4.2	创建 XMLHttpRequest 对象	96
4.2.1	同步用法	97
4.2.2	异步用法	97
4.2.3	readyState 属性	97
4.2.4	XMLHttpRequest 的属性 和方法	98
4.2.5	常见错误	104
4.2.6	更复杂的问题	104
4.3	POST 方法	109
4.4	使用 POST 和 GET 方法的 优缺点	111

4.5 其他 Ajax 技术	112
4.5.1 隐藏框架	112
4.5.2 隐藏的内联框架	119
4.5.3 动态脚本加载	122
4.5.4 图像和 Cookie	125
4.6 本章小结	129
4.7 练习	130
第 5 章 XML 的使用	131
5.1 XML 基础	131
5.1.1 创建标记	131
5.1.2 XML 语法	132
5.1.3 格式良好且有效的 XML	133
5.2 使用 JavaScript 提取 XML 数据	138
5.2.1 使用节点	138
5.2.2 根据名称访问 XML 元素	140
5.2.3 访问属性值	141
5.3 使用 CSS 显示 XML 数据	145
5.3.1 使用 CSS 显示 XML 文档	145
5.3.2 在 Ajax 中使用 CSS	146
5.4 本章小结	147
5.5 练习	147
第 6 章 调试与错误处理	149
6.1 JavaScript 错误处理	149
6.1.1 处理异常	150
6.1.2 onerror 事件处理程序	152
6.1.3 Mozilla JavaScript 控制台	153
6.1.4 Microsoft Script Debugger	155
6.1.5 Firebug	157
6.2 DOM 检查器	159
6.2.1 Firefox DOM 检查器	159
6.2.2 IE DOM Inspector	161
6.2.3 Mouseover DOM Inspector(MODI)	161
6.3 Ajax 故障诊断	162
6.3.1 使用 Firebug 控制台解决 XMLHttpRequest 问题	162
6.3.2 Live HTTP Headers	163
6.3.3 ieHTTPHeaders Explorer Bar	165
6.4 本章小结	165
6.5 练习	166
第 7 章 Web 服务、API 和 Mashup	167
7.1 什么是 Web 服务	168
7.1.1 公共 Web 服务	168
7.1.2 消费第三方 Web 服务	169
7.2 Web 服务的结构	171
7.2.1 REST 方法	172
7.2.2 SOAP 方法	172
7.3 将 Web 服务集成到 Ajax 应用程序	174
7.3.1 使用 XMLHttpRequest 消费服务	174
7.3.2 同源策略	175
7.3.3 创建应用程序代理	175
7.4 使用脚本标记	183
7.5 未来替代方案	185
7.6 使用 API	186
7.7 Web 服务和 API 之间的区别	186
7.8 Google Maps API	187
7.8.1 Google Maps API 密钥	187
7.8.2 Map 对象	188
7.8.3 Geocode	188
7.8.4 XMLHttpRequest 工厂方法	189
7.9 Mashup	197
7.10 Ajax 和 Mashup 的关系	197
7.11 使用 Flickr API	198
7.11.1 标记云(加权清单)	198
7.11.2 使用 Flickr API 密钥	199
7.11.3 创建示例应用程序	199
7.11.4 Flickr 中的地理标记照片	200
7.11.5 显示来自 Flickr 的照片	209
7.12 本章小结	213
7.13 练习	213

第 8 章 XSLT 和 XPath	215	9.4.1 问题	278
8.1 XSLT 及其用途	216	9.4.2 模式	278
8.2 XSLT 元素	217	9.5 拖放列表模式	288
8.2.1 xsl:stylesheet	218	9.5.1 问题	288
8.2.2 xsl:output	218	9.5.2 模式	288
8.2.3 xsl:includes	219	9.6 错误处理模式	302
8.2.4 xsl:template、xsl:apply-templates 和 xsl:call-template	219	9.6.1 问题	303
8.2.5 xsl:if	221	9.6.2 模式	303
8.2.6 xsl:choose	221	9.7 本章小结	308
8.2.7 xsl:for-each	222	9.8 练习	308
8.2.8 xsl:value-of	223	第 10 章 使用外部数据	309
8.2.9 xsl:sort	223	10.1 使用 XML 新闻种子	309
8.2.10 xsl:variable	224	10.1.1 RSS 0.9x	311
8.3 主要浏览器对 XSLT 的支持	224	10.1.2 RSS 2.0	312
8.4 执行一个转换	224	10.1.3 RSS 1.0	314
8.4.1 在 IE 中执行转换	225	10.1.4 Atom	315
8.4.2 在 Firefox 中执行转换	229	10.2 从 XML 种子中提取数据	316
8.4.3 在服务器端执行转换	232	10.2.1 提取 XML 数据	316
8.5 创建购物车的 XSLT 样式表	234	10.2.2 提取字符串数据	323
8.6 Xpath 及其用途	243	10.3 使用 Ajax 构建在线种子 阅读器	324
8.7 Xpath 的基本功能	244	10.4 本章小结	334
8.7.1 XPath 表达式	244	10.5 练习	334
8.7.2 XPath 函数	246	第 11 章 JSON	335
8.8 使用 Xpath 查询 XML 文档	248	11.1 JSON 语法	335
8.9 使用 XSLT 和 Ajax 的购物车 示例	253	11.1.1 数据类型	336
8.10 本章小结	259	11.1.2 对象字面量	336
8.11 练习	260	11.1.3 数组字面量	337
第 9 章 模式	261	11.1.4 使用 JSON 解析器	338
9.1 设计模式背景知识	262	11.2 数据传输格式	339
9.2 表单验证	262	11.3 Ajax 和 JSON	341
9.2.1 问题	262	11.3.1 创建请求	341
9.2.2 模式	262	11.3.2 解析响应	342
9.3 鼠标悬停模式	270	11.3.3 将 JSON 数据添加 到页面	343
9.3.1 问题	270	11.4 在 PHP 中使用 JSON	347
9.3.2 模式	270	11.5 本章小结	349
9.4 轮询服务器模式	278	11.6 练习	349

第 12 章 高级示例：可排序列表	351	12.5 与用户交互：索引页面	363
12.1 使用 MySQL	351	12.6 使用 Ajax 更新	366
12.1.1 创建 MySQL 表	352	12.6.1 创建 POST 请求	368
12.1.2 向表中添加数据	353	12.6.2 创建 GET 请求	369
12.1.3 创建数据库连接	354	12.6.3 结果处理	370
12.2 创建数据库查询	355	12.6.4 添加样式	370
12.2.1 获得当前字段的值	356	12.7 文件	371
12.2.2 排序列表	356	12.8 本章小结	371
12.3 编辑数据库记录	357	附录 A 习题答案	373
12.3.1 插入记录	357	附录 B Ajax 资源：架构和库	397
12.3.2 删除记录	359	附录 C JavaScript 资源	405
12.4 使用 Scriptaculous 实现拖放	360	附录 D JavaScript 语言参考	409
12.4.1 创建可拖放的元素	360		
12.4.2 创建可排序元素	361		

第 1 章

Ajax 简介

在人类漫长的历史长河中，充斥着各种各样的岔路口、数不清的选择和无数的假想。同样，在人类的科技发展征程中，适者生存的法则残酷而又现实。在过去的“战役”中，我们亲眼目睹了 VHS 超越 Betamax；PC 超越微型计算机；Internet Explorer (IE) 超越 Netscape Navigator；更多类似的状况即将会发生在 DVD 格式领域的争斗中。超越并不代表取代，并不代表一种技术一定好于另一种技术，只能说明某种技术在特定的时间内能够满足人们的要求，因而变得日益流行起来。时至今日，我们还是能找到许多痴情于 Betamax 磁带技术的追随者，他们认为这种技术更灵巧、品质更高等。这并不表示这些人的想法是错误的。或许，某种技术的暂时性淘汰让某些人不免有点沮丧和不情愿，但它们毕竟过去也曾辉煌过。

Internet 的进化发展也有自己的岔路口。到目前为止，人们仍在进行一场轰轰烈烈的辩论讨论“胖客户端”与“瘦客户端”孰优孰劣。简单讲，就是选择让浏览器处理大多数工作，还是选择让另一端的服务器处理大多数工作。在最开始的时候，也就是 90 年代中期，“胖客户端”思想似乎马上就要胜出。那个时候，由于 IE 4 和 Netscape Navigator 4 的到来，带来了动态 HTML 技术，该技术使用脚本语言控制页面，从而可以不刷新页面就实现拖放项和隐藏/显示菜单。但是，仅在一年之内，伴随着服务器端技术(例如，ASP 和 PHP)的引入，技术的发展趋势突然移向“瘦客户端”。到目前，客户端技术仍然存在，但当前 Internet 模型和 Web 页面技术还广泛基于服务器端方法，即输入数据、发送页面到服务器、等待响应。

当两种相似技术中的某一种迅速进入主导地位时，人们可能会忘记另一种技术的优点。例如，页面验证的某些方面在浏览器上也可以同样做得很好。如果在电子邮件文本框中输入 fake e-mail，就不需要到服务器端进行验证。JavaScript 也能以同样的效率和更快的速度执行验证。虽然很多人都会客户端和服务端上同时进行验证，但大多数页面都只会尝试在服务器上进行处理。如果 Web 总是出现错误，那么处理速度就会变得很慢。尽管带宽已经提高了 10 倍，但超时、页面未找到、按钮无响应和链接无响应的错误仍然没有消失。因此，用来解决这种反应迟缓的方法正在变得越来越常见。

一些公司已经开始着手重新评估自己当前所正在使用的方法，以确定是否能在多个层面上改善用户的体验——不仅是让页面显示和响应的速度更快，同时也提供更加无缝和丰富的用户体验。这常需要我们回到以前的旧技术中去。使用这种“新”方法来创建 Web 应用程序的第一个也是最好的示例是，Google 的 Gmail 技术。Google 早在 Google Suggest 和 Google Maps 两个应用程序中也使用过此技术，只是这两个应用程序当时的表现并不出色，

所以并不引人注目。而 Windows Live Mail(以前叫 Kahuna)、Amazon 的搜索引擎 A9.com、Yahoo 在线管理图片的 flickr.com 都缺少一种通用的说法来描绘它们的功能,直到 2005 年的一篇网络文章改变了这种情况,在该文章中作者将这些技术统一称为 Ajax。

1.1 什么是 Ajax

Ajax 这一琅琅上口的词汇的作者是 Jesse James Garrett,它第一次出现在该作者 2005 年为 Adaptive Path 公司创作的一篇文章 *Ajax: A New Approach to Web Applications* 中,这篇文章可以在 <http://adaptivepath.com/publications/essays/archives/000385.php> 上找到。如果你还未阅读过该文章,建议您阅读一下(但这最好是在看完本章之后,因为它可能使你在理解 Ajax 时产生偏差)。Ajax 同时还是一个首字母缩略词,基于同样的原因,我们并不马上对其进行介绍。虽然 Ajax 的第一次出现是在该文章中,但文章中所描述的功能其实早已经存在了。

简单讲, Ajax 是一套编程技术,或者说是一种特殊的 Web 编程方法。通过这些编程技术,开发人员能够使用来自服务器的输入无缝地更新 Web 页面或一部分 Web 应用程序,而且这种更新不需要立即刷新页面。这并不是说浏览器就不与 Web 服务器连接。事实上,在最初的那篇文章中,作者描绘的并不完整,因为他没有提到服务器端技术仍要经常用到。而很可能情况是用于描绘网页的页面或数据仍必须通过服务器端来更新。这和 Ajax 技术的不同在于 Ajax 模型中更新页面的位置被改变了。在本章后面,会清楚地看到这两个模型。

Garrett 的文章为我们描绘了这样一个世界, Web 应用程序可以在功能上与 Windows 不相上下,而它的关键字则是“丰富的功能”、“快速的响应”和“简洁的特性”。他把 Ajax 描绘为一种新的应用程序,可以弥补 Windows 和 Web 应用程序之间空缺的新型应用程序,并说明 Gmail、Google Suggest 和 Google Maps 就是这一新方法的代表。

上面提到的那篇文章(甚至“Ajax”这一词汇)将人们分成了两大阵营。虽然很多人还是非常喜欢它,但也有很多开发人员指责它的名称“Ajax”,认为它索然无味,对于文章中描述的技术,也认为毫无新鲜可言,只不过是发挥想象。

但是,多数人也认为被 Jesse James Garrett 所描述的技术和思想所打动(例如,人们声称“如果在从头开始设计 Web 时,不应让用户再苦苦等待”。此外,对于这类应用程序的设计人员,他们的挑战是应当忘记有关 Web 的任何已知的限制,而去设想一个充满无限可能性的世界)。Ajax 让人们通过现有的成熟稳定的方法来创建 Web 应用程序,而不是去使用那些不可靠的 Beta 版技术。它让开发人员可以利用现有的 JavaScript、样式表和文档对象模型技术,而不用夜以继日地追逐最新的基于标记的页面构建语言。它让开发人员从繁忙的工作中解脱出来,甚至一些工作的招聘广告也将条件改为“具备 5 年的 Ajax 工作经验”。

上面说了这么多,我们还不知道 Ajax 到底具体能做什么,像往常一样,最好的方法就是具体看看当前正在 Web 上使用的 Ajax 技术。

1.1.1 Ajax 的应用

毫无疑问, Ajax 的最佳应用典范是 Gmail, 它允许用户编辑和更新电子邮件和收件箱, 而无需不断地刷新页面。对于晚上使用 Outlook、Outlook Express 或 Thunderbird 应用程序的人们, Ajax 技术使他们能够使用基于 Web 的电子邮件系统作用代替。但遗憾的是, Gmail 在不登录的情况下无法演示, 而且其注册只对有限的几个国家开放, 因此在这里只能看看其他几个示例。

1. flickr

Yahoo 的 flickr.com 是一个管理照片的站点, 可以通过它“在线排序、存储、搜索和共享照片”。以前, flickr 使用 Flash 作为其照片显示界面的主要工具, 但在 2005 年 5 月, Yahoo 宣布将采用动态 HTML 和 Ajax 技术 (http://blog.flickr.com/flickrblog/2005/05/from_flash_to_a.html)。必须登录账户才能看到该工具的应用, 因为它是免费的, 并且 Web 应用程序上的照片处理工具是演示 Ajax 技术的最好工具, 所以你应当亲自体验一下。

登录后, 可以通过选择如图 1-1 中所示的 Organize All Your Photos 选项访问该工具。可以将照片拖放到一个批处理中, 然后旋转它们并修改标记。

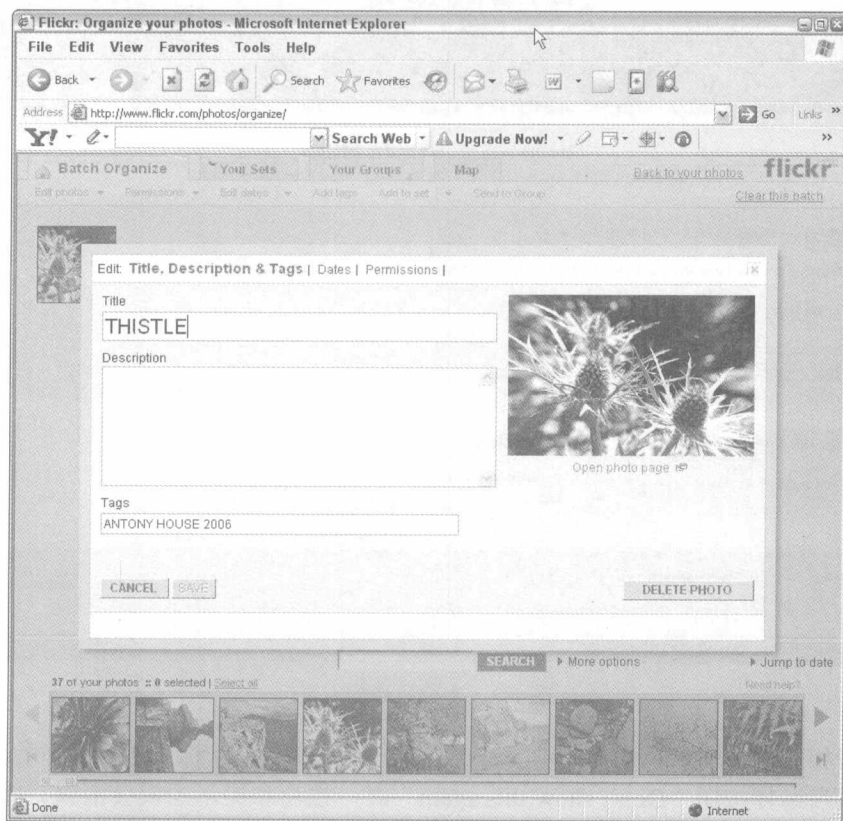


图 1-1 照片管理选项