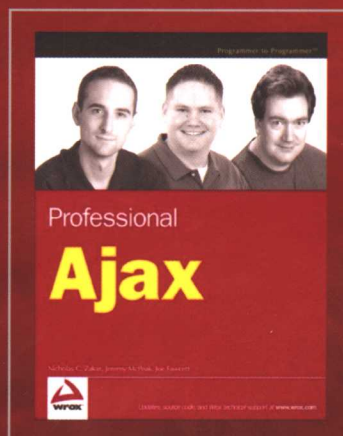


Professional **Ajax**

Ajax 高级程序设计

Nicholas C. Zakas
Jeremy McPeak 著
Joe Fawcett
徐锋 吴兰陟 等译

- Amazon 超级畅销书
- 完整实现 Google Mail 风格的邮件系统
- 实例丰富，使用 PHP、.NET 和 Java 平台



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Ajax高级程序设计

Professional Ajax

Nicholas C. Zakas

Jeremy McPeak

Joe Fawcett

著

徐锋 吴兰陟 等译



人民邮电出版社

POSTS & TELECOM PRESS

图书在版编目 (CIP) 数据

Ajax 高级程序设计 / 扎卡斯, 姆克皮克, 福西特著; 徐锋等译. —北京: 人民邮电出版社, 2006.6

(图灵程序设计丛书)

ISBN 7-115-14867-8

I. A... II. ①扎... ②姆... ③福... ④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 (2006) 第 063836 号

内 容 提 要

本书是关于 Ajax 技术、模式和使用场景的开发人员级教程, 不仅介绍了 Ajax 的基本知识, 还讨论了 Ajax 模式和框架, 同时针对 XML (包括 XPath 和 XSLT)、RSS/Atom、Web 服务、JSON 和 Web 用户界面组件等主题, 阐述了如何将 Ajax 与这些技术有效地结合在一起, 并给出了一个使用 Ajax 开发的 Web Mail 系统完整实例。书中经典的实例、完整的源代码, 都将给读者带来“实战”的指导。

本书内容广泛且深入, 适用于中高层次的 Web 开发人员。

图灵程序设计丛书

Ajax 高级程序设计

◆ 著 Nicholas C. Zakas Jeremy McPeak Joe Fawcett
译 徐 锋 吴兰陟 等
责任编辑 傅志红

◆ 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: 23.75
字数: 534 千字 2006 年 6 月第 1 版
印数: 1-5 000 册 2006 年 6 月北京第 1 次印刷

著作权合同登记号 图字: 01-2006-1463 号

ISBN 7-115-14867-8/TP · 5461

定价: 49.00 元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

版 权 声 明

Original edition, entitled *Professional Ajax* by Nicholas C.Zakas, Jeremy McPeak, Joe Fawcett, published by Wiley Publishing, Inc. Copyright © 2006 by Wiley Publishing, Inc.

All rights reserved. This translation published under license.

The Wrox Brand trade dress is a trademark of Wiley Publishing, Inc. in the United States and/or other countries. Used by permission.

Translation edition published by POSTS & TELECOM PRESS Copyright © 2006 .

本书简体中文版由 Wiley Publishing, Inc.授权人民邮电出版社独家出版。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。

Wrox 商标是 Wiley 出版社在美国及其他国家使用的商标，使用经过许可。

版权所有，侵权必究。

感谢我的家人和 Emily，在过去的这些年里，你们的爱和支持对我而言是无价的。

——Nicholas C. Zakas

感谢我的父母——Sheila 和 William，是你们的熏陶使我喜爱上了阅读。谢谢！

——Jeremy McPeak

译者序

在笔者翻译《Ajax 修炼之道》接近尾声时，dlee（《Ajax 实战》的译者）向我推荐了这本 *Professional Ajax*。在《Ajax 修炼之道》一书的翻译过程中，我深切地感受到“程序员修炼之道”系列书籍一贯的写作风格，语言简洁、明快，知识系统、完整，通过深入浅出的语言很容易就可以让一个初学者完成 Ajax 内功心法的修炼。

但作为一本仅有 300 页左右的薄书，《Ajax 修炼之道》不可能包括更多的实现细节，因此要想进一步地在实战中提高自己的技艺，还需要一些“武功招式”，否则仅有深厚的内力也不足以“临场杀敌”。稍加翻阅 *Professional Ajax* 一书之后，我就确定这正是一本讲述“武功招式”的好书，因此就决定趁势继续开始本书的翻译。

在本书的前 2 章中，作者以简洁、明快的风格阐述了 Ajax 的演化历史和技术前身，以及 Ajax 最基本的使用方法，使得本书自成体系，即便是初学者也能够很快开始自己的 Ajax 学习历程。第 3 章则将 Ajax 的应用场景抽象成为通信控制和失效处理模式，对 Ajax 针对不同需求的解决方案提供了完整的参考。

接着，在第 4~8 章中，分别针对 XML（包括 XPath 和 XSLT）、RSS/Atom、Web 服务、JSON 以及 Web 用户界面组件等五个主题，阐述了如何将 Ajax 和这些技术有效地结合在一起，经典的实例、完整的源代码，都将给读者带来“实战”的指导。在此之后，本书讲述了一个利用 Ajax 开发的 Web Mail 系统实例（第 9 章 AjaxMail），来说明如何通过 Ajax 及相关技术达到 Gmail 所实现的用户体验。

最后，本书详细地讲述了 JSPspan、DWR 和 Ajax.NET 三种 Ajax 框架的使用方法，对于 PHP、JSP 和 ASP.NET 的 Web 开发人员，都可以找到适合自己的框架。

在翻译本书的过程中，原作者文思流畅、代码完整，这都给笔者留下了深刻的印象。本书也是笔者翻译的多本书籍中感到最为畅快的一本。如果一定要指出它的缺点，那么可能只有一点，即在大部分章节中，文中例子使用的是现在使用率稍低于 Java 和 .NET 的 PHP。不过，Ajax 关注的是前端技术，后台代码的实现并非本书的重点，加上 PHP 代码的易读性，并不会给读者带来太多的困难。笔者所在的开发团队就基于这些例子，简单地修改成了 ASP.NET 的实现版本，整个过程没有遇到太多的周折。

Ajax 作为一项热点技术，近期相关的书籍也层出不穷，作为《Ajax 修炼之道》和本书的译

者，加上对《Ajax 实战》和《Ajax 基础教程》的阅读体会，在此与大家分享一下对这四本书籍的看法：

- 如果你想快速地了解 Ajax 技术，可以选择《Ajax 基础教程》和《Ajax 修炼之道》；
- 如果你想迅速提高自己动手实现 Ajax 解决方案，建议看《Ajax 基础教程》和本书；
- 如果你想用最短的时间系统化地了解 Ajax 的方方面面，《Ajax 修炼之道》是首选；
- 如果你想深入研究 Ajax 技术，那么《Ajax 实战》是不可错过的好书。

最后，在此向图灵公司的傅志红编辑和技术上的好友李锟（dlee）先生表示衷心的感谢，他们的支持和指导是本书顺利完成的基石；感谢在本书翻译过程中进行了初译的吴兰陟、陈绍继、郑端和胡冰，他们为本书的翻译提供了许多帮助。最后还要感谢母亲多年来的教导，以及妻子许高芳一直以来的默默支持与鼓励。

鉴于笔者水平有限，译文中难免有不妥和疏漏之处，恳请读者批评指正，也欢迎读者来信（xf@csai.cn）与我一起交流 Ajax、Web 2.0、B/S 软件开发相关的更多话题。

徐 锋

2006 年 5 月于厦门紫荆园

preface 前言

随着近来 JavaScript 的发展，Web 开发人员能够在 Web 应用程序中创建出史无前例的用户体验。现在 Web 开发人员可以摆脱原来具有统治地位的“点击-等待”模式，利用一种名为 Ajax 的技术，就可以将原本在桌面应用程序中才可以实现的功能带到 Web 中。

Ajax 是一个涵盖很广的术语，它指的是，通过 JavaScript 发出异步 HTTP 请求，在不重载页面的情况下从服务器获取信息。这些请求可以通过多种方法执行，使用多种不同的数据传输格式。将这种远程数据获取能力与文档对象模型（DOM）的交互性结合，就产生了新一代的 Web 应用程序，它看起来违反了所有传统的 Web 应用程序规则。诸如 Google、Yahoo! 和微软这样的大公司都投入了专门的资源来研发，其目标是创建视觉效果和行为与桌面应用程序一样的 Web 应用程序。

本书讨论了 Ajax 的方方面面，包括向服务器发出 HTTP 请求的不同方法，在往返通信中承载数据时使用的不同格式。读者还将了解一些在网站和 Web 应用程序中执行客户端-服务器通信的各种 Ajax 技术与模式。

读者对象

本书的目标读者有两类：

- ❑ 想提高网站和应用程序可用性的 Web 应用程序开发人员；
- ❑ 想进一步深入了解 JavaScript 语言的中级开发人员。

此外，本书也非常适合已经熟练掌握了以下相关技术的读者：

- ❑ XML
- ❑ XSLT
- ❑ Web Services
- ❑ PHP
- ❑ C#
- ❑ HTML

□ CSS

本书不适用于对于上述技术缺乏基本了解的初学者。同样，很好地掌握 JavaScript 对于理解本书至关重要。如果读者还不熟悉这些知识，应先阅读 *Professional JavaScript for Web Developers* (Wiley 出版社, ISBN 0-7645-7908-8)¹。

本书内容

本书是一本关于 Ajax 技术、模式和使用场景的开发人员教程。

本书从考查 Ajax 的起源开始，阐述了 Web 的演化过程以及直接促使 Ajax 产生的各种新技术，并详细地讨论了 Ajax 所包含的帧、JavaScript、cookie、XML 以及 XMLHttpRequest 等知识。

在此之后，本书将转到特定 Ajax 技术的实现上。对隐藏帧、动态 iframe 和 XMLHttpRequest 等请求代理技术进行了比较，说明了什么时候应该使用哪种方法。为了阐述得更加清晰，其中包含了一个关于 HTTP 请求和响应的简要描述。

在讲完了不同请求类型的基本原理后，本书提供了一些更深入的实例，说明在网站或应用程序中什么时候和如何使用 Ajax。本书还阐述了不同数据传输格式（包括纯文本、HTML、XML 和 JSON）的优点和缺点，同时介绍了 Web 服务，并说明如何用它来实现 Ajax 技术。

本书的最后一部分，将带你一起创建一个完整的名为 Ajaxmail 的基于 Ajax 的 Web 应用程序，它融入了本书前面介绍的大量技术，另外还将介绍几个用来简化 Ajax 通信开发的 Ajax 库。

组织结构

在开始实际的实现之前，本书首先讲述与 Ajax 起源相关的背景知识。接下来，讨论了实现客户端-服务器通信的不同方法，为本书的其余部分打下基础。我们建议读者按顺序阅读本书，因为每一章都建立在以前各章的基础上。

各章的内容如下所示：

- 第 1 章：“什么是 Ajax”。本章阐述了 Ajax 的起源及其相关的技术。它介绍了 Ajax 随 Web 发展而发展的过程，以及各种术语和技术的创造者。
- 第 2 章：“Ajax 基础”。本章介绍了实现 Ajax 通信的不同方法，包括隐藏帧技术和 XMLHttpRequest。本章还讨论了每种方法的优缺点，同时也给出了何时使用它的原则。
- 第 3 章：“Ajax 模式”。本章主要讨论使用 Ajax 的设计模式。网站和应用程序使用 Ajax 的方法有很多种，这些方法已经组织成一系列模式，描述了纳入 Ajax 的最佳实践。
- 第 4 章：“XML、XPath 和 XSLT”。本章介绍了 XML、XPath 和 XSLT 等技术，作为

1. 本书即将由人民邮电出版社出版。——编辑注

Ajax 的补充。讨论的核心放在使用 XML 作为数据传输格式，使用 XPath 和 XSLT 来访问和显示信息。

- ❑ 第 5 章：“基于 RSS/Atom 的聚合”。本章讨论的是将 Ajax 与数据聚合格式 RSS 和 Atom 结合在一起，创建一个基于 Web 的新闻聚合器（news aggregator）。
- ❑ 第 6 章：“Web 服务”。本章将 Web 服务带入到 Ajax 中，从实例解释了如何从客户端调用 Web 服务，以及如何创建一个服务器端代理，使其可以在浏览器的安全约束下工作。
- ❑ 第 7 章：“JSON”。本章介绍了 JavaScript 对象符号（JSON），它是在 Ajax 通信中另一种可选的数据传输格式。本章阐述了它与 XML、纯文本相比的优点与缺点。
- ❑ 第 8 章：“网站的用户界面组件”。本章则使用前几章中介绍的技术来创建基于 Ajax 的能够包含在网站中的用户界面组件。
- ❑ 第 9 章：“AjaxMail”。本章带你一起开发一个名为 AjaxMail 的完整 Web 应用程序。该应用程序是一个基于 Ajax 的电子邮件系统，使用了前面描述的许多技术。
- ❑ 第 10 章：“Ajax 框架”。本章阐述了三个 Ajax 框架：支持 PHP 的 JPSPAN，支持 Java 和 JSP 的 DWR，以及支持 .NET 框架的 Ajax.NET。这些框架的目的都是使 Ajax 开发过程的某个部分自动化。

运行环境

要运行本书中的例子，对软件环境的要求是：

- ❑ Windows 2000、Windows Server 2003、Windows XP 或 Mac OS X 操作系统；
- ❑ IE 5.5 及以上版本（Windows）、Mozilla 1.0 及以上版本（所有平台）、Opera 7.5 及以上版本（所有平台）或 Safari 1.2 及以上版本（Mac OS X）。

这些例子的完整代码可以从图灵网站中本书配套网页下载。

排版约定

为了使读者从本书中获得更多信息，并明了发生了什么，在本书中使用了一些约定的表示法。

这样的文本框中的都是重要的、不应忘记的信息，它们与周边的文本信息直接相关。

与当前阐述的内容相关的技巧、提示、窍门，将以楷体字表示。

至于正文的风格则包括：

- ❑ 当引入新术语和重要的词时会采用楷体字来强调。

- 键盘输入的格式类似于: Ctrl+A。
- 对于文件名、URL 和文中的代码,使用的字体类似于: persistence.properties。
- 对于代码的表示则采用了两种方法:

在代码实例中,带灰色背景的代码说明是强调的、新的或重要的代码。
不使用灰色背景的代码,说明该代码对上下文而言不重要,或者前面已经显示了。

源代码

如果你想研究本书中的代码,可以自己手动输入代码,也可以直接使用本书附带的源代码。本书中使用的所有源代码都可以在 www.wrox.com 网站上下载¹。在该网站中,搜索本书的标题(通过搜索框或标题列表),然后在本书的详细页面中点击“Download Code”(下载代码)链接,以获取本书的源代码。

由于书名相近的书很多,因此通过 ISBN 来搜索会更容易找到。本书的 ISBN 是 0-471-77778-1。当下载了代码后,用你喜欢的压缩工具软件将其解压出来。另外你也可以到 Wrox 的代码下载页 www.wrox.com/dynamic/books/download.aspx 中寻找本书的代码,在那里还可以找到 Wrox 出版的所有书籍的配套代码。

勘误表

我们尽了最大努力,以确保文字和代码中没有错误。但是,没有任何东西是完美的,都会发生错误。如果你在本书中找到错误,例如拼写错误、不完善的代码片断,请向我们反馈,我们将万分感激。通过递交勘误表,可以节省其他读者被其困扰所花费的时间,同时也可以帮我们进一步提高本书质量。

要寻找本书的勘误页面,首先登录 www.wrox.com,通过搜索框或标题列表找到本书。然后,在本书的详细页面中点击“Book Errata”(勘误表)链接。在该页面中你将看到由 Wrox 编辑发表的,已经提交的勘误表。在 www.wrox.com/misc-pages/booklist.shtml 的完整图书列表中,针对每本书的勘误表都有一个相应的链接。

如果在勘误页面中没有发现你找到的错误,请访问 www.wrox.com/contact/techsupport.shtml,填写上面的表单,将你发现的错误发给我们。我们将会及时地检查;如果正确,那么就会在本书的勘误页面上发表消息来修改这个错误,并且在本书后续版本中纠正该错误。

1. 本书的源代码可以链接图灵网站 (www.turingbook.com) 中本书页面的“资源”下载。——编者注

p2p.wrox.com

如果要与作者及同行进行讨论，可以参加位于 p2p.wrox.com 的 P2P 论坛。该论坛是一个基于 Web 的系统，你可以发送与 Wrox 书籍和有关技术相关的贴子，并且与其他读者和技术用户进行交流。该论坛提供了订阅功能，如果你选中了感兴趣的话题，那么当论坛中该贴子有回复时，就会自动通过电子邮件发给你。Wrox 的作者、编辑、业内专家以及读者都会在这个论坛中出现。

在 <http://p2p.wrox.com> 中，你会发现大量不同的论坛，它不仅可以帮助你阅读本书，同时也可以帮助你开发自己的应用程序。要加入该论坛，其步骤为：

- (1) 打开 p2p.wrox.com，然后点击“Register”（注册）链接；
- (2) 阅读用户条款，然后点击“Agree”（同意）；
- (3) 填写加入论坛所需的信息以及愿意提供的可选信息，然后点击“Submit”（提交）按钮；
- (4) 你将收到一封电子邮件，其中的信息告诉你如何激活你的账户，以完成整个注册过程。

没注册时可以阅读论坛中的贴子，但要回复或发表贴子，就一定要注册。

当加入该论坛后，就可以发表新贴子，也可以回应其他人的贴子。你可以阅读任何时候的贴子。如果你希望某个特定论坛的新贴子能够发电子邮件给你，只需点击该论坛列表中论坛名称后面的“Subscribe”（订阅）图标就可以实现。

要了解关于 Wrox P2P 的详细使用信息，一定别忘了阅读 P2P 的 FAQ，在这个 FAQ 中回答了许多与论坛软件相关的问题，同时也包含了许多关于 P2P 和 Wrox 书籍的常见问题。要阅读这个 FAQ，只需点击 P2P 页面上的 FAQ 链接。

acknowledgments 致 谢

写 这样一本书相关的人有许多，我们必须在此对本书做出贡献的人们表示感谢。

首先也是最重要的是，感谢 Wiley 出版社的每个人及他们的支持：Jim Minatel 启动了该过程，Gabrielle Nabi 对前几章的编写提供了帮助，当 Gabrielle 离开后，John Sleeva 接着完成了他的工作。同样，我们也要对技术编辑 Alexei Gorkov 深表感谢，他为保持本书内容的可信做出了神奇的贡献。

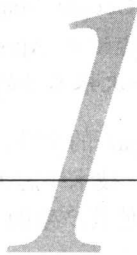
最后，要感谢所有在本书出版前提出宝贵建议的人们，他们包括 Martin Honnen、Peter Frueh、Mike Shaffer、Brad Neuberg、Steven Peterson 和 Eric Miraglia。

contents 目 录

第 1 章 什么是 Ajax	1	2.4 小结	42
1.1 Ajax 的诞生	1	第 3 章 Ajax 模式	43
1.2 Web 的演化过程	2	3.1 通信控制模式	43
1.2.1 JavaScript	2	3.1.1 预先获取	43
1.2.2 帧	2	3.1.2 页面预载入实例	44
1.2.3 隐藏帧技术	3	3.1.3 提交节流	51
1.2.4 动态 HTML 和 DOM	3	3.1.4 表单增量验证实例	53
1.2.5 iframe	3	3.1.5 字段增量验证实例	59
1.2.6 XMLHttpRequest	4	3.1.6 定期刷新	62
1.3 真正的 Ajax	4	3.1.7 新评论提示实例	63
1.4 Ajax 原则	5	3.1.8 多阶段下载	67
1.5 Ajax 背后的技术	6	3.1.9 附加信息链接实例	68
1.6 谁在使用 Ajax	7	3.2 失效处理模式	70
1.6.1 Google Suggest	7	3.2.1 取消待处理的请求	70
1.6.2 Gmail	8	3.2.2 重试	73
1.6.3 Google Maps	8	3.3 小结	74
1.6.4 A9	9	第 4 章 XML、XPath 和 XSLT	75
1.6.5 Yahoo!News	10	4.1 浏览器对 XML 的支持	75
1.6.6 Bitflux Blog	11	4.1.1 IE 中的 XML DOM	75
1.7 混淆与争议	11	4.1.2 Firefox 中的 XML DOM	84
1.8 小结	12	4.1.3 跨浏览器兼容的 XML	88
第 2 章 Ajax 基础	13	4.1.4 基本的 XML 实例	89
2.1 HTTP 基础	13	4.2 浏览器对 XPath 的支持	95
2.1.1 HTTP 请求	13	4.2.1 XPath 概述	95
2.1.2 HTTP 响应	15	4.2.2 IE 中的 XPath	96
2.2 Ajax 通信技术	16	4.2.3 使用命名空间	97
2.2.1 隐藏帧技术	17	4.2.4 Firefox 中的 XPath	100
2.2.2 XMLHttpRequest 请求	30	4.2.5 使用命名空间解析器	101
2.3 进一步考虑	40	4.2.6 跨浏览器兼容的 XPath	102
2.3.1 同源策略	40	4.3 浏览器对 XSLT 的支持	103
2.3.2 缓存控制	41	4.3.1 XSLT 概述	103

4.3.2 IE 中的 XSLT	106	6.7 小结	176
4.3.3 Firefox 中的 XSLT	110	第 7 章 JSON	177
4.3.4 跨浏览器的 XSLT	112	7.1 什么是 JSON	177
4.3.5 重访“最佳选择”	112	7.1.1 数组字面量	177
4.4 小结	115	7.1.2 对象字面量	178
第 5 章 基于 RSS/Atom 的聚合	116	7.1.3 混合字面量	179
5.1 RSS	116	7.1.4 JSON 语法	180
5.1.1 RSS 0.91	117	7.1.5 JSON 编码和解码	180
5.1.2 RSS 1.0	118	7.2 JSON 与 XML	181
5.1.3 RSS 2.0	119	7.3 服务器端 JSON 工具	182
5.2 Atom	119	7.3.1 JSON-PHP	183
5.3 FooReader.NET	120	7.3.2 其他工具	184
5.3.1 客户端组件	120	7.4 创建自动提示的文本框	185
5.3.2 服务器端组件	130	7.4.1 功能概述	185
5.3.3 连接客户端和服务端	136	7.4.2 HTML	186
5.4 安装	143	7.4.3 数据库表	188
5.5 测试	145	7.4.4 架构	188
5.6 小结	146	7.4.5 类	189
第 6 章 Web 服务	147	7.4.6 自动提示控件	190
6.1 相关技术	147	7.4.7 建议提供者	205
6.1.1 SOAP	147	7.4.8 服务器端组件	207
6.1.2 WSDL	150	7.4.9 客户端组件	208
6.1.3 REST	153	7.5 小结	210
6.2 .NET 连接	156	第 8 章 网站的用户界面组件	211
6.3 设计决策	156	8.1 创建滚动新闻用户界面组件	211
6.4 创建 Windows 平台的 Web 服务	157	8.1.1 服务器端组件	212
6.4.1 系统需求	158	8.1.2 客户端组件	213
6.4.2 配置 IIS	158	8.1.3 设置新闻样式	221
6.4.3 编写 Web 服务	160	8.1.4 实现滚动新闻用户界面组件	222
6.4.4 创建程序集	161	8.2 创建天气用户界面组件	223
6.5 Web 服务与 Ajax	164	8.2.1 Weather.com 提供的 SDK	223
6.5.1 创建测试工具	164	8.2.2 服务器端组件	224
6.5.2 IE 使用的方法	166	8.2.3 客户端组件	232
6.5.3 Mozilla 使用的方法	167	8.2.4 从服务器获取数据	233
6.5.4 通用方法	170	8.2.5 定制天气用户界面组件	234
6.6 跨域的 Web 服务	172	8.2.6 实现天气用户界面组件	237
6.6.1 Google Web API 服务	172	8.3 创建 Web 搜索用户界面组件	238
6.6.2 创建代理	173	8.3.1 服务器端组件	238

8.3.2 客户端组件.....	239	9.6 小结.....	326
8.3.3 定制 Web 搜索用户界面组件.....	244	第 10 章 Ajax 框架.....	327
8.3.4 实现 Web 搜索用户界面组件.....	247	10.1 JPSpan.....	328
8.4 创建网站搜索用户界面组件.....	248	10.1.1 工作原理.....	328
8.4.1 服务器端组件.....	249	10.1.2 安装 JPSpan.....	328
8.4.2 客户端组件.....	254	10.1.3 创建服务器端页面.....	328
8.4.3 定制网站搜索用户界面组件.....	259	10.1.4 创建客户端页面.....	334
8.4.4 实现网站搜索用户界面组件.....	260	10.1.5 错误处理.....	338
8.5 小结.....	262	10.1.6 类型转换.....	339
第 9 章 AjaxMail.....	263	10.1.7 JPSpan 小结.....	340
9.1 需求.....	263	10.2 DWR.....	341
9.2 架构.....	264	10.2.1 工作原理.....	341
9.2.1 使用的资源.....	264	10.2.2 安装 DWR.....	341
9.2.2 数据库表.....	264	10.2.3 创建客户端页面.....	344
9.2.3 配置文件.....	266	10.2.4 使用自定义类.....	345
9.2.4 AjaxMailbox 类.....	267	10.2.5 错误处理.....	349
9.2.5 执行操作.....	289	10.2.6 dwr.xml 中的更多配置项.....	349
9.3 用户界面.....	294	10.2.7 转换器.....	351
9.3.1 文件夹视图.....	297	10.2.8 DWR 小结.....	352
9.3.2 阅读视图.....	299	10.3 Ajax.NET.....	352
9.3.3 写邮件视图.....	301	10.3.1 工作原理.....	352
9.3.4 布局.....	303	10.3.2 安装 Ajax.NET.....	352
9.4 整合.....	303	10.3.3 创建网页.....	354
9.4.1 辅助函数.....	305	10.3.4 简单类型和复杂类型.....	360
9.4.2 mailbox 对象.....	306	10.3.5 会话状态.....	361
9.4.3 回调函数.....	323	10.3.6 Ajax.NET 小结.....	363
9.4.4 事件处理函数.....	325	10.4 小结.....	363
9.5 最后一步.....	325		



在 2001~2005 年间，万维网（WWW）涌现出了大量新技术和新方法论，它们赋予了这种曾经静态的媒体新的生命。在线产品手册和产品目录已不再是主流，而 Web 应用程序开始成为在线服务的主要形式。与传统网站不同，Web 应用是为用户提供即时服务。无论是为了业务流程管理还是个人兴趣，随着用户对更加丰富功能的期望，开发人员都将转而创建新的交互模式。

这些不太知名、很少使用的技术在 Web 浏览器中已经存在了一段时间了，它们使 Web 迈出了一大步，彻底推翻了传统的使用模式，当访问新数据或新的应用程序业务逻辑时不再需要重新载入整个页面。各个公司都开始尝试在网页中实现部分内容的动态载入，只向客户端传送一小部分数据，以得到更快速、更好的用户体验。

处于这一运动热潮最前沿的当数 Google 公司。当搜索引擎被广泛接受之后，Google 的工程师们就开始在网站中名为 Google Labs 的栏目上展示新的尝试。在 Google Labs 中的许多项目，诸如 Google Suggest、Google Maps，都只有单个网页，虽然从不卸载页面，但数据不断更新。这些革新着眼于将桌面软件用户界面的各种特色（affordance）迁移到浏览器屏幕范围之中，它们受到了广泛赞誉，引领了 Web 开发新时代的到来。

无数开源软件 and 商业化产品都开始考虑利用这种新 Web 应用程序模型进行开发。这些项目使用 JavaScript 远程调用、Web 远程过程调用、动态更新等各种不同的术语来阐述该技术。但是很快，一个新的术语应运而生了。

1.1 Ajax 的诞生

2005 年 2 月，Adaptive Path 公司的 Jesse James Garrett 在网上发表了一篇名为《Ajax：一种 Web 应用程序开发的新方法》的文章（现在还可以在 www.adaptivepath.com/publications/essays/archives/000385.php 看到）。在这篇文章中，Garrett 阐述了他为什么认为 Web 应用程序正在填平与传统桌面应用程序之间的鸿沟。他引用了一些新的技术，并以几个 Google 的项目作为例子，说明了如何将传统的、基于桌面应用程序的用户交互模型应用到 Web 上。然后他说出了两句引起人们大量兴趣、兴奋和争论的话：