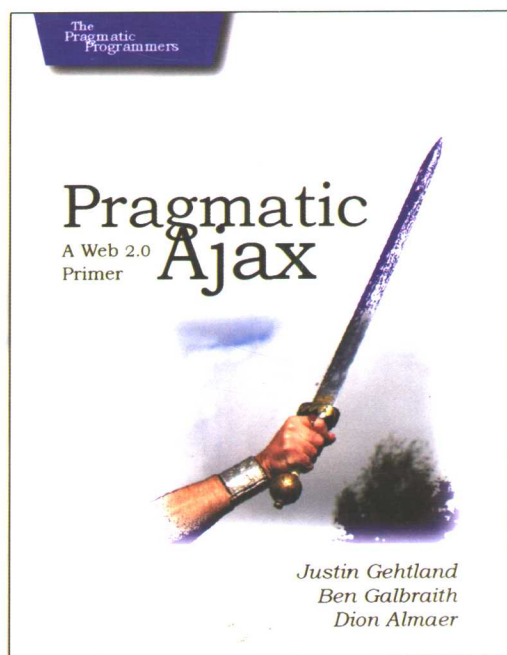


Ajax 修炼之道

Pragmatic Ajax

— Web 2.0 入门

A Web 2.0 Primer



[美] Justin Gehrtland

Ben Galbraith

Dion Almaer

徐锋 胡冰

译 著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Pragmatic Ajax^A

TP393.09

84

Ajax 修炼之道

— Web 2.0 入门 —

TP393.09
84

[美] Justin Gehtland
Ben Galbraith 著
Dion Almaer

徐 锋 胡 冰 译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

原书作者是 Ajaxian.com 的创始人, JavaOne、TheServerSide 等诸多高级别会议的演讲者。本书作为 Pragmatic 系列之一, 从实践出发, 通过实例展示 Ajax 的诸多特性, 手把手教你实现 Google Map 的绚丽效果。不仅教会你 Ajax 的技术细节, 同时还带你了解各种功能强大的主流 Ajax 工具包 (Dojo、Prototype、Script.aculo.us、DWR、Backbase、SmartClient、Ajax.NET、SAJAX、JSON-RPC), 掌握 Ajax 实时查看源代码的方法以及进行代码调试的相应方法, 学习 Ajax 的开发模式和框架。

本书后半部分共用了五个章节, 从服务器端编程的角度详细讲述了 Ajax 同 PHP、Ruby On Rails、Java 和 .NET 等语言的融合, 最后还介绍了与 Ajax 相类似的其他 RIA 技术以及 Ajax 的精彩未来 (E4X、Canvas、SVG 的相关应用)。本书的全部代码下载地址: <http://media.pragprog.com/titles/ajax/code/ajax-code.zip>。

本书秉承了《程序员修炼之道》的特点——“从小工到专家”, 各个层次的 Web 开发人员都能从本书中获益匪浅。

0-9766940-8-5 Pragmatic Ajax by Justin Gehrtland, Ben Galbraith, Dion Almaer.

All rights reserved. Authorized translation from the English language edition published by The Pragmatic Programmer, LLC..

本书简体中文专有翻译出版版权由 The Pragmatic Programmer, LLC. 授予电子工业出版社。未经许可, 不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号: 图字: 01-2006-2133

图书在版编目 (CIP) 数据

Ajax 修炼之道: Web 2.0 入门 / (美)哥特兰 (Gehtl and, J.), (美)高伯瑞斯 (Galbraith, B.), (美)艾米亚 (Almaer, D.) 著; 徐锋, 胡冰译. —北京: 电子工业出版社, 2006.4

书名原文: Pragmatic Ajax: A Web 2.0 Primer

ISBN 7-121-02458-6

I. A... II. ①哥...②高...③艾...④徐...⑤胡... 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 030775 号

责任编辑: 周 筠

印 刷: 北京智力达印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×980 1/16 印张: 19.5 字数: 290 千字

印 次: 2006 年 4 月第 1 次印刷

定 价: 29.80 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077。质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

译者序

长期奋战在开发一线的我们，不久前一直饱受一个问题的困扰，那就是用户对 B/S 体系结构下的用户界面颇有微词。虽然我们采用的是 Visual Studio .NET，在用户界面的友好性和互动性方面已经有了很大的改观，但受限于 Web 的静态特性及“刷新”的本质模式，原来在 C/S 体系结构下能够手到拈来的效果，要在 B/S 中实现却是难上加难。

但 Google Mail、Google Map、Google Suggest 等这些应用是怎么实现的呢？偶然的的机会，发现其奥秘居然蕴含于“阿贾克斯”！什么，这怎么和足球挂上了钩了呢？噢，原来是名为 AJAX 的技术。通过一段时间的研究，发现它实际上是“新瓶装旧酒”，主要是基于 JavaScript，通过 XMLHttpRequest 来实现异步的远程通信，并利用 XML 来实现数据的封装、更新等，最终达到交互性更强的用户界面效果。

通过在项目中的一些试用之后，发现 AJAX 的确给 Web Application 的开发提供了一个强大的支撑工具，它使得开发出来的系统可用性大大得到提高，用户满意的笑容让我们感到无比的欣慰。而正在这时，武汉博文的周筠女士将这本书推荐给我，并且告知 CSDN 的孟岩对该书给予了很高的评价，希望我能够同步进行本书的翻译工作。一方面，我对 AJAX 的兴趣正浓；另一方面，翻译国外作者还正在写作中的书还是头一回，这也使我更有跃跃欲试的冲动。

而当我收到已完成的部分章节之后，更是被其中的内容深深地吸引：全书贯穿了一个真实、简单、有效的实例，首先从原理出发讲述了 AJAX 的基础知识，然后逐一介绍了针对 PHP、Ruby on Rail、Spring、.NET 等不同的服务端技术的 AJAX 开发库。通过阅读本书，读者不仅能够对 AJAX 的基础技术有所了解，还能够宏观地认识 AJAX 技术，选择相应的工具，并且能够实际的动手使用 AJAX，真所谓是研习 AJAX 的“修炼之道”。

基于原来对 AJAX 的了解与应用，再加上本书的研读与翻译，笔者有许

多自己的感悟与体会，在此借用一个隐喻来阐释。综观人类的战争史，存在“轻灵”与“重装甲”交替主宰战争的现象：盔甲战胜了布衣、轻骑兵又战胜了盔甲，坦克战胜了轻骑兵、导弹兵又战胜了坦克。而在计算机应用史上，从最早的哑终端发展到胖 C/S 模式，又从胖 C/S 发展到瘦 B/S 模式，现在又开始倡导胖 B/S 模式……因此对于一项技术，成败的关键都在于“在正确的地方正确地应用”。

我在此向武汉博文的周筠女士、CSDN 的孟岩先生致谢，感谢你们对我的信任与支持；感谢魏泉、刘铁锋两位编辑，你们的工作使本书的译文更加准确、到位；感谢长年来一起合作共事的胡冰、吴兰陟，你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，因此在此必须感谢每一位读者，希望能够得到您的反馈与批评，欢迎来信（xf@csai.cn）与我一起交流与 AJAX、Web 2.0、B/S 软件开发的更多话题。

徐 锋

2006 年 3 月于厦门紫荆园

目 录

Contents

第 1 章 基于 Ajax 构建富 Internet 应用	1
1.1 故事三篇	2
1.2 Google Maps: The Missing Spark	4
1.3 什么是 Ajax	5
1.4 现在要往何处去	8
第 2 章 构建 Google Maps	9
2.1 技术专家	10
2.2 你自己的 Google Maps	11
2.3 构建 Ajax 化地图	16
2.4 结论	47
第 3 章 Ajax 实战	49
3.1 将 Web 应用 Ajax 化 (Ajaxifying)	49
3.2 用 Ajax 解决	49
3.3 令人生厌的细节	57
3.4 总结	60
第 4 章 解析 Ajax	61
4.1 客户端的 JavaScript 技术概述	62
4.2 操作 Web 页面	68
4.3 查询数据	74
4.4 总结	77
第 5 章 Ajax 框架	79
5.1 框架、工具包和程序库	79
5.2 基于 Dojo 工具包的远程调用	84

5.3 基于 Prototype 程序库的远程调用	92
5.4 总结	94
第 6 章 Ajax UI, 第一部分	95
6.1 针对 UI 的 Ajax 和 JavaScript	95
6.2 总结	123
第 7 章 Ajax UI, 第二部分	125
7.1 一些标准用法	125
7.2 并非全是酒和玫瑰	140
7.3 总结	149
第 8 章 Ajax 应用系统调试	151
8.1 查看源文件	151
8.2 DOM Inspector (查看器)	152
8.3 调试 JavaScript	163
8.4 总结	173
第 9 章 可降解的 Ajax	175
9.1 什么是可降解的 Ajax?	175
9.2 确保 Ajax 应用系统可降解	177
9.3 总结	188
第 10 章 JSON 和 JSON-RPC	189
10.1 JSON-RPC	192
第 11 章 服务端框架集成	197
11.1 不同的集成策略	198
第 12 章 基于 PHP 的 Ajax	201
12.1 PHP 框架	201
12.2 使用 Sajax	202
12.3 XOAD	210
12.4 总结	215
第 13 章 基于 Rails 的 Ajax	217
13.1 Ruby on Rails	217

13.2	Ajax 集成	221
13.3	Ajax in Rails 的未来	234
第 14 章	基于 DWR 的代理型 Ajax	237
14.1	DWR	238
14.2	结论	252
第 15 章	ASP.NET 和 Atlas	253
15.1	BorgWorX	254
15.2	Atlas	256
15.3	总结	265
第 16 章	Ajax 的未来与超越	267
16.1	数据操作	267
16.2	UI 操作	271
16.3	预测	283
16.4	总结	286
索引		287

第 1 章

基于 Ajax 构建富 Internet 应用

Building Rich Internet Applications with Ajax

这是一本关于如何开发高效Web应用的书籍。我们不打算让大家对此话题感到疑惑。实际上在本书的主题中，所包含的内容都是诸如XHTML、JavaScript、CSS等已经出现了十余年的成熟技术。我们不但承认这些技术的有效性，而且还将充分地利用它们。虽然这些标准已经存在了一段时间，但并不意味着不能在此之外构建一些新的、令人兴奋的东西。技术，就像布丁甜点一样，要融入可口和满意总是需要一些时间的。

Ajax（及Web 2.0）就是利用这些成熟的Internet标准化技术，构建一个切实可行的应用开发平台。它将这些稳定的标准技术组合在一起，形成了一个更易于理解、更统一的版本，使其价值比迄今为止各个部分的价值之和还大。基于Ajax，你将可能实现两个终极目标¹：功能齐备（feature-filled）的用户界面，以及无干扰（no-hassle）、零安装（no-install）的部署模式。

Jesse James Garrett发明Ajax这个术语已经很久了。当他第一次在公共场合发表这个术语时，采用了“异步JavaScript和XML”（*Asynchronous JavaScript And XML*），然后就像SOAP一样，用其首字母组成了一个新词。不过，这是一个非常有力的词²。基于这个词，Jesse James能够套上更“胖”的、免安装的Web应用这一产业级的发展趋势，并赋予更多关注的焦点。

为某一事物命名是十分有效的。但即便这样，也并不是足以使其成为趋势，

¹ 译者注：原文是 Holy Grail，即圣杯。是一个宗教术语，传统观点认为它是一只圣餐杯，曾经盛放基督的血液。在美国人的使用习惯中，会用它来表示伟大的使命、终极的目标，是一个神秘不可达的东西。

² 译者注：AJAX 的发音很容易让人想起荷甲的“阿贾克斯”，因此使其流行得更加迅速。

它仍只是星星之火，还存在一些不足。如果没有真实的例子往往会使得说服力不足。下面则是一个开发团队的真实故事，它描述了如何转而使用我们所说的Web应用开发方法。

1.1 故事三篇

Hector是一个Web应用开发公司的项目经理。他在Perl CGI、ASP、Servlet和JSP等方面具有丰富的开发经验，但也遇到了一些麻烦。去年他的团队为一个财富500强的公司开发一套CRM应用，该公司在全世界都有办事处。原来使用的是一个绿屏的哑终端/主机应用³；而该公司希望利用Internet的优势，快速地将应用部署到每个办公室。

Hector和他的团队将大部分的精力花应用系统的服务端。他们使用了来自Java社群的一种MVC框架来开发业务逻辑；用一种高性能的持久化框架来访问数据库；利用基于消息的基础结构来连接其他的原有系统。

■ 昨天

在客户端开发方面，Hector和他的团队都对CSS很精通。Web页面似乎能够随心所欲；客户需要圆形收口（rounded corner）就能够做出圆形收口。改变颜色？也很简单。多种色彩模式？没问题。但事实上，Hector和他的团队早已开始对其用户界面感到担忧了。大家都知道，Web的运作模式只有一种：其本质就是分发静态页面。当用户需要更多数据时，就会导致一个彻底的界面刷新。从效率的角度来看，这并不是一个最佳的选择，但这就是现在Web工作的模式，用户也早已习以为常了。

而就在几周以前，Hector的客户和他开了一次会。这些客户都很有教养、彬彬有礼。他们了解Web，也知道Internet访问会带来的限制。事实上，在这次会议前，Hector从未看到过他们生气。

但客户代表一走进会议室，开发团队就感觉到发生了什么事。他拿着笔记本，而在此前从未有过。他吼着走进房间，开发团队的所有人都面面相觑：我们干了什么？客户坐在桌子边上，打开笔记本，埋头敲了一会键盘。在他

³ 译者注：在哑终端/主机的计算模式中，终端用户使用的是一种非智能显示终端，通常是绿底白字的。

边敲键盘时，边告诉开发团队：“昨天晚上，我和我妻子受邀到CEO家里参加了一个party。”“噢”，开发团队心里想，“肯定不是什么好事情。”

“我当然是欣然接受这个机会，”他继续说到。“我还从没有参加过呢。这个项目让我进入了他的视野。”（“哦⁴”，Hector 越发感到惊讶）。“当我无法从城市地图中找到去那里的路时，我就连上 Internet。我发现了**这个！**”他说到最后一个词时，话音中充满着埋怨与嘲笑。他转了一下笔记本电脑，使得周围的人都能够看到。在他浏览器窗口中呈现出的，就是 Google Maps。“这是怎么做的，”他咬着牙说，“我们能够做到这种效果吗？”

■ 今天

自从那次会议之后，Hector 和他的开发团队被推到了一个必须重新构思用户界面的境地。Hector 开始研究 Google 是如何彻底地摆脱了传统的思维习惯，开发出这样的应用。他找到了一篇 Jesse James Garrett 写的文章，将这种方法称为 Ajax。从那开始，他就开始研究所有与这种新型 Internet 应用构建方法相关的东西。

开发团队接着就着手重新实现 UI。他们使用 JavaScript 和 DHTML 技术来提供更多动态的体验。特别重要的是，他们充分发挥了一个在现代浏览器中拥有的、特别有用的 XMLHttpRequest（简称 XHR）对象的优势。它使得 Hector 和他的开发团队能够实现，在无需重新刷新页面中所有信息的情况下，发出请求并从服务器上接收新的数据。

换句话说，Hector 成为了从 Web1.0 跨越到 Web2.0 的先锋。他的客户再次感到了高兴。

■ 明天

Hector 接下来做了什么呢？他的团队学习了大量的 JavaScript、XHTML 等相关知识，甚至还更深入地学习了 CSS。其结果确实令开发团队感到兴奋：用户体验确实看起来和应用程序没有什么区别了，现在除了不需要做一个安装程序之外，其他都和应用程序类似。但他们却隐隐感到事情有些不妙。

现在，他们编写了大量的 JavaScript 代码。这使得所有的页面都需要处理，并且完成 XHR 存取需要大量实际的友善（real, honest-to-goodness）的代码。尽管 JavaScript **看起来**很像 Java，但仍然发现它们存在着大量的不同。

⁴ 译者注：此处原作者用了“Double Uh oh”来表示其越来越感到惊讶。

现在他们需要管理、测试和维护两种代码。

因此, Hector 决定停下来去寻找解决这些问题的方法。他发现大部分 Web 应用开发框架已经快速地将 Ajax 工具纳入其中。很快, Hector 和他的开发团队就发现了 Tapestry 组件, Spring 标签库、ASP.NET widget (窗口小部件)、Rails 助手 (Rails helper) 及 PHP 库等等, 利用它们根本无需其他工作就可以有效地获得 Ajax 的优势。在不久的将来, Ajax 开发将完全无缝地集成于这些框架中。这也恰好是 Hector 所需要的。

1.2 Google Maps: The Missing Spark

Google Maps (<http://maps.google.com>) 的确点燃了 Ajax 的火种。而这本不应该由 Google 来实现。首先我们看看 Google 以前给我们留下的感觉: 最好的搜索引擎, 全世界最简单的 UI。它只是一个底色全白的页面, 正中间有一个文本框和一个按钮。不可能再有比这个更加简单的界面了。如果 Google 是一首曲子, 那必然是 Philip Glass⁵ 的作品。

Google 进入在线地图服务领域之后, 我们才发现实际上我们需要一种更简单、干扰更少的方法来查阅地图。而且我们完全可以实现这一目标, 不必将它抛在一边。Google 巧妙地利用了 XHR 回调, 提供了在第一张页内可滚动的地图。如果你想查看地图的其他部分, Google 只是将原来的移出屏幕并查询出新的。在此没有混乱的页面刷新; 大量无变化的文本也并未重载。特别是, 在刷新地图时不需等待大量的数据刷新, 地图应用就应该这样做。

然后当我们点击了推针 (push pin) 时就会出现一个信息提示框, 里面有一些文本信息, 并带有阴影。它是一个时代结束的标志。Hector 的故事在我们的身边一次又一次地发生。某人的老板或客户、同事看到 Google Maps 时就会说: “为什么我们做不到?”

作为程序员, 另一个反应就是: “我希望能够去开发这类应用系统。”这也意味着 Google Maps 以及类似的应用系统给人的印象是, 这是高科技, 它需要特殊的团队、特殊的开发人员才能够做到。本书无论如何也要消除这种观点。正如第9页中的第2章“构建 Google Maps”所示, 使 Web 页面“又歌

⁵ 译者注: 菲利普·格拉斯 (Philip Glass) 是著名的极微主义乐派 (即简约派) 中最知名的代表人物, 也二十世纪最有影响力的作曲家之一。

又舞”起来并非需要你了解一种新的工具。甚至给你印象更深的是，Google Map并非完整的Ajax应用；它没有应用任何现行的异步回调技术，实际只是动态HTML而已。

1.3 什么是 Ajax

我们很难用通俗的话来阐释Ajax。其中原因有两个方面：

- Ajax 可以看作是一组技术。
- Ajax 可以看作是一种体系结构。

■ Ajax：异步 JavaScript 和 XML

“Ajax”这一名字源于已有应用技术的集合：JavaScript、XML，以及浏览器和服务端之间的异步通信通道。当对其定义时，包含的内容是：

- 普通内容的表示，使用 XHTML 和 CSS。
- 动态显示和交互，使用文档对象模型（DOM，Document Object Model）。
- 数据交换与操作，使用 XML 和 XSLT。
- 异步数据查询，使用 XMLHttpRequest 或 XMLHttpRequest（来自于微软）。
- 用 JavaScript 将这些绑定在一起。

尽管它是基于这些已有技术的开发方法，但也很快会使其带来的麻烦要比产生的利益更多。因此本书的目的就是告诉你如何实现：

- 融合 Ajax 化（Ajaxian）技术来实现数据传输，而不使用规格化的 XML。
- 跳过 DOM API，操作存储于内存中的页模型。
- 使用异步方式调用服务器，虽然它很强大但也很危险。
- 将 XMLHttpRequest 的复杂性抽象出来。

正是因为这些理由，因此 Ajax 更重要的定义是：

■ Ajax：一种体系结构

Ajax最令人兴奋的发展趋势是改变架构Web应用的方式。让我们先看看“古老”的Web应用架构方式：

1. 为应用系统中的每一个事件定义一个页面：查看、购买、结帐等。
2. 每个事件或操作，都将向浏览器返回一个完整的页面。
3. 这些页面将呈现给最终用户。

这些东西直到现在看起来都觉得十分自然。这种认知在 Web 出现之初就形成了，当时 Web 的确不是应用系统。Web 的起源更类似于文档库；它指的是能够简单地在文档间连接与跳转的特殊方法。它更关注文档、数据共享，没有太深含义的交互。

让我们先将注意力转到胖客户端的桌面应用程序上一会。假设每一次点击后，在应用程序屏幕上的所有组件都会重绘一次，这种情形你敢想象吗。这听起来就是胡言呓语，不是吗？而在 Web 领域，直到 Ajax 出现之前，却一直是这样的。

Ajax 创建了一种新的体系结构。这种体系结构的最核心部分如下。

- 较少的服务端事件：现在在 Web 应用中组件可以发出较小的请求，获取少量信息，然后通过 DOM 变换更新部分页面，而非刷新整个页面。
- 异步：向服务器提交请求之后，不会影响浏览器继续操作。用户可以继续使用应用系统的其他部分，而 UI 会更新以提醒用户请求已经发出。
- onAnything：我们能够根据用户的操作向服务器回传指令。现代浏览器能够捕获绝大部分操作系统确认的用户事件：单击鼠标、鼠标移过、按下键盘等。任何用户事件都能够触发一个异步的请求。

图 1.1 展示了一个 Ajax 页面的新生命周期：

1. 用户通过指定 URL 来发起最初的请求。
2. 服务器返回原始的 HTML 页面。
3. 浏览器通过内存中的 DOM 树来展现页面。

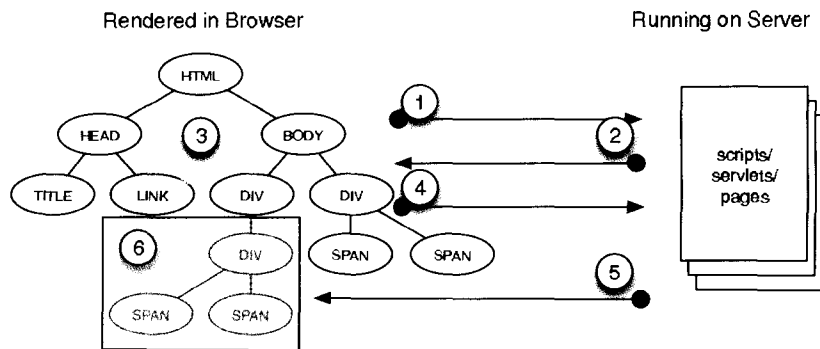


图1.1 Ajax页面生存周期

4. 用户行为引发基于其他 URL 发出后续的异步请求，其内容不在现有的 DOM 树中。
5. 浏览器响应原页面中的回调函数，并返回数据。
6. 浏览器解析结果，并使用新的数据更新内存中的 DOM 树，然后再呈现在用户屏幕上（该页重绘了，但没有“刷新”）。

这听起来很美妙，不是吗？但对于这个变化也必须小心。对于 Web 而言，最重要的事件之一是任何人都可以使用它。它的迅猛发展得益于其简单的语法。如果我们做过头，那么用户可能不得不面对一个全新的 UI 风格。这也是 Flash UI 受到诟病的主要原因，用户必须面对新的符号、隐喻及必须的操作才能够达到可用的结果。可用性是一个很重要的话题，我们将会在第 7 章“Ajax UI 第二部分”（第 125 页）中更为深入地探讨。

■ Ajax：未来

Ajax 的目标是什么？希望获得什么样的未来？这是一个至关重要的问题，这是因为 Ajax 看上去还是一个意义随着上下文不同而不同的未定型术语。Ajax 本身是描述一组技术的统一术语。我相信作为术语本身，随着其统

一和长期的发展，最终可能在下一年代消失于大家的意识中。

这是因为在本书中学到的技术，最终将成为你所喜爱的Web应用开发平台的底层技术基础。它将不再是闪着星光的小道具、精巧的窍门，而仅仅是Web应用工作的方式。那岂不是意味着本身毫无价值？远非如此。现在你需要知道如何做出这种效果，将来你需要知道如何调试Web应用。只不过将来这类应用不再称为Ajax，而仅仅是Web应用。这的确是件好事。

1.4 现在要往何处去

本书的剩余部分将带你领略Ajax的全景。我们将剖析这种新的应用范型转变，并深入地研究Ajax背后的支撑技术。还将介绍大部分可用的工具集和框架，它们使得应用这些技术更为简单，就像加入一行代码似的。并将会看到如何在你所喜欢的平台上发挥这种新的开发方式的优势，并集成到自己的应用中。

更重要的是，我们还将讨论如何更有效、实用地使用Ajax。因为只要一步错就可能会南辕北辙，就像当火车离站时就开到错误的轨道上一样。我们希望本书成为在这一新的、快速发展的“风景区”观光的指南。将帮助你找到使Ajax有助于你项目的方法。我们并不想推销什么（除了这本书）。但我们坚信Ajax是一个很重要的发展，希望能够帮助你最好地掌握和利用它。

让我们从点起 Ajax 火花的 Google Maps 开始。

第 2 章

构建 Google Maps

Creating Google Maps

对于我们大多数人来说，Google Maps(<http://maps.google.com>)点燃了 Ajax 技术革命。然而早在 Google Maps 出现以前，Ajax 技术就已经缓缓成为主流的 Web 技术，而在最近的记忆当中还没有什么技术在常用的浏览器中提供了如此印象深刻的体验。而 Google Maps 展示出的的是一个蕴藏于已知技术中的巨大潜在空间。

正如我们在第 1 章中所说的，Ajax 最初定义是 XMLHttpRequest 对象加上使用 XML 更新 DOM 树。但是，Ajax（及 Web 2.0）现在定义的范围就广得多。本章将说明 Google Maps 的技术基础，以及现代的基于浏览器的应用系统如何只基于标准的 HTML 和 JavaScript 实现全新的 Web 应用。

本章的目的是完全揭示令大家叫绝的 Google Maps 所采用的技术。我们将发现这里是如此的迷人和重要，以至于你流连于此，不愿离开。如果真的是这样，你可以先跳过本章去阅读其他章节，然后再回到本章，不用担心，我们对此没有任何意见。

本章包含大量的代码。它们都可以在线获得，因此可以包含本书所有源码的压缩包¹。如果你阅读的是本书的 PDF 版本，则只需点击该链接就可以获得该文件。可是，如果这些文件包含 HTML，那么可能会在你的浏览器中无法正确呈现。但如果只是想看运行的应用程序是没问题的。如果你想阅读代码，请使用浏览器的“查看源代码”选项。

¹ 原文注：下载链接：<http://pragmaticprogrammer.com/titles/ajax/code.html>。