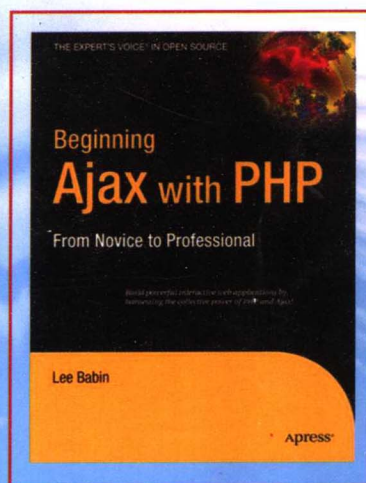


Beginning Ajax with PHP

Ajax与PHP 基础教程

[加] Lee Babin 著
徐 锋 陈绍继 译

- 为 PHP 程序员量身打造的 Ajax 教程
- 注重实践，内含大量实战代码
- 凝聚业界最佳实践





图灵程序设计丛书

Web开发系列

Ajax 与 PHP 基础教程

Beginning Ajax with PHP

[加] Lee Babin 著

徐 锋 陈绍继 译

人民邮电出版社

北 京 ,

图书在版编目 (CIP) 数据

Ajax 与 PHP 基础教程 / (加) 巴班 (Babin, L.) 著; 徐锋, 陈绍继译.

—北京: 人民邮电出版社, 2007.9

(图灵程序设计丛书)

ISBN 978-7-115-16252-6

I. A... II. ①巴...②徐...③陈... III. ①计算机网络—程序设计—教材②PHP 语言—程序设计—教材 IV. TP393.09 TP312

中国版本图书馆 CIP 数据核字 (2007) 第 071333 号

内 容 提 要

本书是介绍如何将 Ajax 和 PHP 这两种流行技术结合起来创建新一代 Web 应用的图书。它讲述了日常的 Web 应用程序开发任务, 演示了如何为表单验证、文件上传监控、数据库驱动信息显示和处理、Web 服务、Google Maps 集成等搭建 PHP/Ajax 解决方案。

本书适合于各个层次的 Web 开发人员。

图灵程序设计丛书

Ajax 与 PHP 基础教程

-
- ◆ 著 [加] Lee Babin
 - 译 徐 锋 陈绍继
 - 责任编辑 陈兴璐
 - ◆ 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>
 - 三河市海波印务有限公司印刷
 - 新华书店总店北京发行所经销
 - ◆ 开本: 800 × 1000 1/16
 - 印张: 12.5
 - 字数: 296 千字 2007 年 9 月第 1 版
 - 印数: 1—5 000 册 2007 年 9 月河北第 1 次印刷

著作权合同登记号 图字: 01-2006-7030 号

ISBN 978-7-115-16252-6/TP

定价: 39.00 元

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

前言

做技术工作是件很有趣的事情，因为每次陷入困境的时候，总会涌现出新方案让我既惊又喜。Web开发就是这样的典型例子。例如，最初我们都必须与基本的HTML打交道，除了表格比较麻烦之外，其他都很简单。但很快地，简单的Web网站开始变成了复杂的Web应用程序，因此诸如PHP之类的脚本语言就必须了解了。服务器端开发成为主流已有很长时间了，而诸如CSS和XHTML之类的Web标准将成为Web世界的下一代主流。

在Ajax出现之后，开发人员再一次发现自己面对艰难的选择。不过和以往一样，未来将会怎样发展我们都不知道，但它最终将结束Web的“单击，等待”模式。只要有安装浏览器的计算机和因特网连接，Ajax就能使用户获得类似桌面应用程序那样的用户界面。现在，开发人员也更倾向于将应用程序移植到用户灵活性更大的环境中去。

但不管Ajax如何承诺，许多Web开发人员都对必须学习JavaScript（一种关键的Ajax技术）感到恐惧。不必担心！本书将向PHP用户展示如何把Ajax融合到自己的Web应用中，并且无需陷入JavaScript语法中，全书将通过实际例子和操作来介绍各个主题。全书共分为14章，每章的主要内容如下：

第1章“Ajax概述”，介绍Ajax技术的背景，解释它之所以成为当前最热门的Web开发技术的详细原因。

第2章“Ajax基础”，介绍无论何种应用程序都将涉及的Ajax基本语法及概念。

第3章“PHP和Ajax”，通过几个示例说明了客户端和服务端如何整合在一起，构建出非常棒的Web应用程序。

第4章“数据库驱动的Ajax”，在前面几章的基础上引入MySQL。

第5章“表单”，解释Ajax如何通过诸如实时表单验证等任务来改善用户体验。

第6章“图像”，展示如何通过Ajax来上传、操作和显示图像。

第7章“一个真实的Ajax应用程序”，将根据前面学到的知识构建一个基于Ajax的图库程序。

第8章“人性化的显示”，介绍了几种在构建RIA（富因特网应用）时始终会用到的最佳实践。

第9章“Web服务”，介绍如何将Ajax与Web服务相结合，从而更有效地将诸如Google和Amazon之类厂商所提供的内容整合到自己的网站中。

第10章“包含空间信息的Web应用”，介绍一个Ajax实现的标志性应用程序：Google Maps API。

第11章“跨浏览器问题”，讨论为目前广泛使用的多种浏览器开发Ajax应用时需要注意的问题。

第12章“安全”，介绍一些由Ajax集成引发的几个攻击因素，并说明如何避免。

第13章“测试与调试”，介绍一些能方便JavaScript调试的工具。

第14章“DOM”，介绍Ajax驱动的应用程序中最关键的元素：DOM（文档对象模型）。

联系作者

读者可以通过lee@babinplanet.ca与作者联系。

致谢

完成本书并不轻松，它需要很多人的帮助和建议。本书的写作过程也不例外。

首先，我必须感谢才华横溢、认真负责的技术专家Quentin Zervaas。Quentin一直为保证本书质量而无私地贡献他的时间及辛勤的工作。他不辞辛苦地努力使每个代码片段、概念都尽可能完美。而且在本书的写作最困难的一段时间内，Quentin发挥了重要作用。总之，没有他的帮助本书是不可能完成的。谢谢Quentin，你的协助使我顺利渡过最困难的时期。

可能你会认为本书是由作者自己一个人完成的，其实对于任何一本书，都有许多关键的参与者在确保其按进度、高质量地完成，本书也不例外。我要感谢管理本书并保证本书定稿的Jason Gilmore和Richard Dal Porto。他们给予了我巨大的帮助，我要对他们自始至终的耐心和理解表示衷心的感谢。

我还要感谢我深爱的妻子Dianne。我在长时间地工作以至于无法陪伴她，而她却一直毫无怨言。在整个写作过程中，她始终支持着我，如果没有她长期的耐心、爱、支持和信心，本书肯定无法完成。

最后我还要谢谢你，本书的读者。我保证这是肺腑之言，真地感谢你拿起这本书（或在电脑中阅读）。如果没有阅读者，毫无疑问我根本没有写作的动力。再次谢谢你的支持，我也真心希望你能够喜欢本书并对你有所助益。

目 录

第 1 章 Ajax 概述	1	4.5 将基于 Ajax 的数据库查询应用到工作中	46
1.1 从 CGI 到 Flash 再到 DHTML	2	4.6 更完善的自动完成功能	48
1.2 当前 Web 应用程序环境的优缺点	2	4.7 载入日历	50
1.2.1 走近 Ajax	3	4.8 小结	51
1.2.2 Ajax 的必要条件	7	第 5 章 表单	53
1.3 小结	7	5.1 引入 Ajax: GET 与 POST	54
第 2 章 Ajax 基础	8	5.2 值的传递	54
2.1 HTTP 请求和响应基础	8	5.3 表单验证	64
2.2 XMLHttpRequest 对象	9	5.4 小结	69
2.2.1 XMLHttpRequest 方法	10	第 6 章 图像	70
2.2.2 XMLHttpRequest 属性	11	6.1 图像上传	70
2.2.3 跨浏览器用法	12	6.2 图像显示	73
2.2.4 向服务器发送请求	13	6.3 图像载入	75
2.2.5 一个简单的 Ajax 示例	14	6.4 略缩图动态生成	76
2.3 小结	18	6.5 小结	80
第 3 章 PHP 和 Ajax	19	第 7 章 一个真实的 Ajax 应用程序	81
3.1 为什么选择 PHP 和 Ajax	19	7.1 程序代码	81
3.2 客户端驱动通信, 服务器端完成处理	20	7.2 应用程序的界面外观	90
3.3 简单示例	20	7.3 工作原理	92
3.3.1 内容缩放	20	7.4 小结	99
3.3.2 自动完成	25	第 8 章 人性化的显示	100
3.3.3 表单验证	32	8.1 何时使用 Ajax	100
3.3.4 工具提示	35	8.1.1 后退按钮问题	101
3.4 小结	38	8.1.2 Ajax 导航	102
第 4 章 数据库驱动的 Ajax	39	8.1.3 隐藏与显示	103
4.1 MySQL 简介	40	8.2 PEAR 简介	104
4.2 连接到 MySQL	41	8.3 HTML_Table	105
4.3 查询 MySQL 数据库	42	8.4 小结	109
4.4 MySQL 技巧和警告	45		

第 9 章 Web 服务	110	12.3.3 GET 和 POST	160
9.1 SOAP Web 服务概述	111	12.3.4 意外的 CSRF 攻击	160
9.1.1 引入 Ajax	112	12.4 拒绝服务	161
9.1.2 开始编码	112	12.4.1 策略 1: 通过延迟来对请求 进行节流	161
9.1.3 SOAP 应用程序的工作原理	116	12.4.2 策略 2: 优化 Ajax 响应数据	162
9.2 小结	120	12.5 知识产权与业务逻辑的保护	164
第 10 章 包含空间信息的 Web 应用	121	12.5.1 策略 1: 对 JavaScript 代码 混淆处理	165
10.1 Google Maps 如此流行	121	12.5.2 策略 2: 实时的服务器端 处理	165
10.2 蓄势待发	122	12.6 小结	168
10.3 该地图应用系统的工作原理	132	第 13 章 测试与调试	169
10.4 小结	143	13.1 JavaScript 错误报告	169
第 11 章 跨浏览器问题	144	13.2 Firefox 扩展	171
11.1 Ajax 的可移植性	144	13.2.1 Web 开发人员工具条	171
11.2 保留后退按钮	146	13.2.2 DOM 解析器	171
11.3 Ajax 响应问题	148	13.2.3 LiveHTTPHeaders	172
11.4 弱化 JavaScript	150	13.2.4 JavaScript 调试器 Venkman	174
11.5 浏览器升级	152	13.2.5 HTML 验证	175
11.6 小结	152	13.3 IE 扩展	176
第 12 章 安全	153	13.3.1 IE 开发人员工具条	176
12.1 增加了受攻击面	153	13.3.2 Fiddler	176
12.1.1 策略 1: 将相关的进入点放到 相同的脚本中	154	13.4 小结	177
12.1.2 策略 2: 采用标准函数来使用 和处理用户输入	154	第 14 章 DOM	178
12.2 跨站脚本	155	14.1 访问 DOM 元素	178
12.2.1 策略 1: 从输入数据中去除 多余的标签	156	14.1.1 document.getElementById	178
12.2.2 策略 2: 在显示客户端提交 的数据时对标签进行转义	157	14.1.2 getElementsByTagName	179
12.2.3 策略 3: 保护会话数据	157	14.1.3 访问表单中的元素	179
12.3 跨站请求伪造	158	14.2 DOM 元素的添加和删除	180
12.3.1 对重要的操作采用一次性 令牌	158	14.3 操纵 DOM 元素	181
12.3.2 通过用户密码来确认重要 的操作	160	14.4 用 DOM 操纵 XML	182
		14.5 将 Ajax 和 XML 与 DOM 结合	183
		14.6 Ajax 位置信息管理器的工作原理	187
		14.7 小结	192

因特网技术的发展一直非常迅速。虽然最早出现的时候显示的是文本信息（因为那时存储空间和内存都很有限），但近几年它已迅速发展成为图形化的、功能很强的媒体。随着其发展，就需要有相关的工具来开发、制造和维护。随着迅速发展的技术，开发人员不断拓展所能实现的功能范围，也开始要求开发工具更加可靠。

为了满足这种需求，出现了大量“Web开发人员”能够使用的工具。诸如HTML、PHP、ASP和JavaScript等语言，可以帮助开发人员在因特网上创建和部署自己的作品。每种语言都已经有多年的发展历史，并给现在的Web开发人员提供了强大的工具集。虽然这些工具的功能日益强大，但在因特网应用程序和更根深蒂固的桌面应用程序之间仍然存在几个主要差异。

在可见的差异中，可能最明显的就是页面请求。在Web应用程序中要执行某些操作，就要向服务器端发送调用请求。为此页面必须刷新，以呈现出从服务器端传给客户端（通常是诸如Firefox或IE等Web浏览器）的最新信息。这不是某个浏览器的缺点，而是所有Web浏览器的HTTP请求/响应协议就是按这种方式工作的（见图1-1）。虽然从理论上讲，这种方式可能工作得很好，但是开发人员开始寻求更简易的方法，以使Web应用程序的响应时间更短。

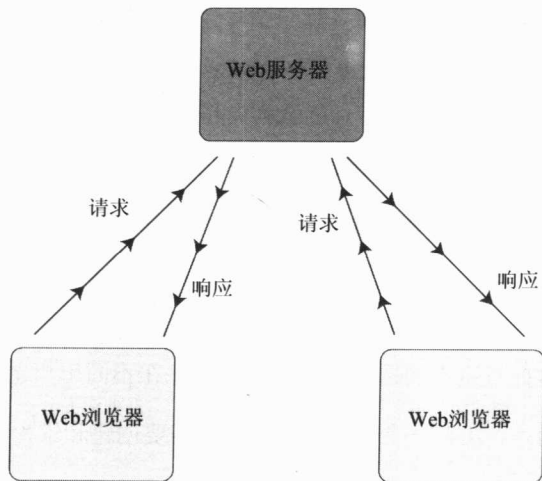


图1-1 因特网当前绝大多数网站所采用的请求/响应方法

1.1 从 CGI 到 Flash 再到 DHTML

开发社区提出需求,商业公司就会给予解决方案。开发工具在设计完成之后,就肯定会有褒贬不一的评价。也许第一种脚本语言就是为了使Web应用设计更加自由,从而避开服务器端CGI(公共网关接口)。

随着CGI的出现,开发人员可以完成一些复杂的操作,如动态图像创建、数据库管理、复杂计算以及动态Web内容创建等。我们现在对Web应用程序的各种认识最早就是源于CGI。不幸的是,CGI虽然解决了很多问题,但其交互和响应的无缝结合仍是个难题。

为了创建真实生动的Web内容,Macromedia公司¹发布了一个功能强大的Flash套件,在那时这一举动可以称得上是“石破天惊”。直到今天Flash仍然实至名归。它使Web开发人员能够构建出可视化的、令人印象深刻的“影片”,并作为网站、应用程序的一部分功能。这些网站比其他网站更“炫”,因为可以在各种浏览器上实现“运动”式的效果。

在专业设计师手下,嵌入Flash的Web网站在视觉效果方面会给人留下深刻印象。同样,在专业开发人员手中,它们可以表现出强大功能。不过在单个处理过程中同时融合视觉设计和开发技能是很少见的;因此Flash应用程序分为两类,一类是视觉效果好但功能少,另一类则是功能很强但界面效果有待提高。而且这种两难境地又夹杂了兼容性问题:为了实现Flash功能,浏览器必须安装一个插件。

还有一种可视化动态技术——DHTML(动态超文本标记语言),虽然已经出现多年但用户群并不大。DHTML是JavaScript和HTML的紧密结合体,其本质就是将HTML和CSS元素与JavaScript结合,在Web浏览器上实现动态的效果。虽然DHTML在JavaScript专业开发人员手里能够实现一些令人印象深刻的效果,但因为需要掌握一些专家级技术,因此并未进入主流²。

如下拉菜单、图像翻转、弹出的工具提示等脚本都很普通,但只是由少数高手开发,大多数开发人员仅仅是应用和部署而已。很少有人自己从头开发这些软件包,因此直到最近也并没有很多人认为JavaScript是一种因特网的强大工具。

1.2 当前 Web 应用程序环境的优缺点

创建基于因特网的Web应用程序有很明显的优缺点。桌面应用程序不断与跨平台兼容问题做斗争,经常要根据完全不同的规则进行编码,而因特网应用程序在不同浏览器中移植则简单得多。加之只有少数主流浏览器拥有巨大的用户基础,因而程序在不同用户之间部署相对比较稳定。

对于在线应用程序而言,还有一个很大的优势是只需要创建和维护一套代码。如果创建的是

1. 已被Adobe公司并购。——编者注

2. DHTML的衰落很大程度上是当时浏览器不兼容造成的。——编者注

桌面应用程序，那么在部署一个修复bug的补丁时，用户要么重新安装整个软件包，要么获取该补丁并安装它。此外，很难判断哪些安装将受到影响。

另一方面，Web应用程序可以只部署在服务器上就能够让所有用户访问。对其功能的任何改变或改进都只需发布到一个中心位置并马上生效。随着开发人员开发能力的增强，他们倾向于开发和维护更高级的产品。

不过任何事情都是有代价的。从维护的角度来看，在中心服务器上部署应用程序是很好的选择，但也将引发新的问题：客户端需要一种访问这个中心服务器的途径。因特网为此提供了一个良好的选择，但速度马上成了新的问题。

例如在使用Word时，单击一个按钮会立即启动某个操作并马上获得响应，而基于因特网的应用程序则需要首先与它连接。虽然高速因特网越来越普及，但很多用户仍然在使用56 Kbps（甚至更低）的调制解调器。因此即使在服务器上能够快速处理信息，发送给最终用户还是需要很长时间。

与该问题相伴的是，对于每个服务器的响应还需要刷新页面，这样就给最终用户带来了很大的挫折感。因此Web应用程序还需要获得桌面应用程序那样的传输速度。如前所述，Flash提供了这样一种手段，并且还提供了强大的ActionScript语言来扩展它，但这需要一个专业人员才能有效地应用它。DHTML通过使用JavaScript也提供了这样的手段，但其代码是很受限制的。

更糟糕的是还经常需要处理浏览器与标准不兼容的问题（甚至完全不遵循标准）。幸运的是，对于这些问题已经有了解决方法——Ajax。这是Jesse James Garrett给“异步JavaScript和XML”所起的名字，它因Google的Gmail等Web应用程序而广泛流行。Ajax基于更流畅的页面载入向服务器端发送请求，并且很少需要刷新整个页面。

1.2.1 走近 Ajax

Ajax在因特网世界引起了大家的重视，不仅仅在于其易于使用且功能强大，还在于它能够吸引几乎所有开发人员的注意。提出Ajax的这两年来，没有出现任何形式的标准（而且彻底基于Ajax来构建其核心的网站还不算太多），但现在Ajax看起来已经像图像翻转一样平常了。

一些一鸣惊人的Web应用程序完全基于Ajax功能。它们不仅巧妙地应用了这一技术，还将Web产业带入一个新的时代，使得标准的Web浏览器能够具备很强的功能，甚至能与桌面应用程序媲美。

例如Flickr（www.flickr.com）或Gmail（www.gmail.com）（参见图1-2），从表面上看它们所提供的服务并没有什么新颖之处（在因特网上有多少网络相册系统和Web邮件服务呀！）。为什么这两个应用程序能够获得如此多（特别是在网络社区中）的关注呢？

我认为这些基于Ajax的新应用程序之所以流行，不是因为它提供了什么新的、令人惊讶的功

能，而是它们以一种有效的、人性化的形式（直到现在，这些在因特网应用程序中还是比较匮乏的）将信息和功能展现给我们。

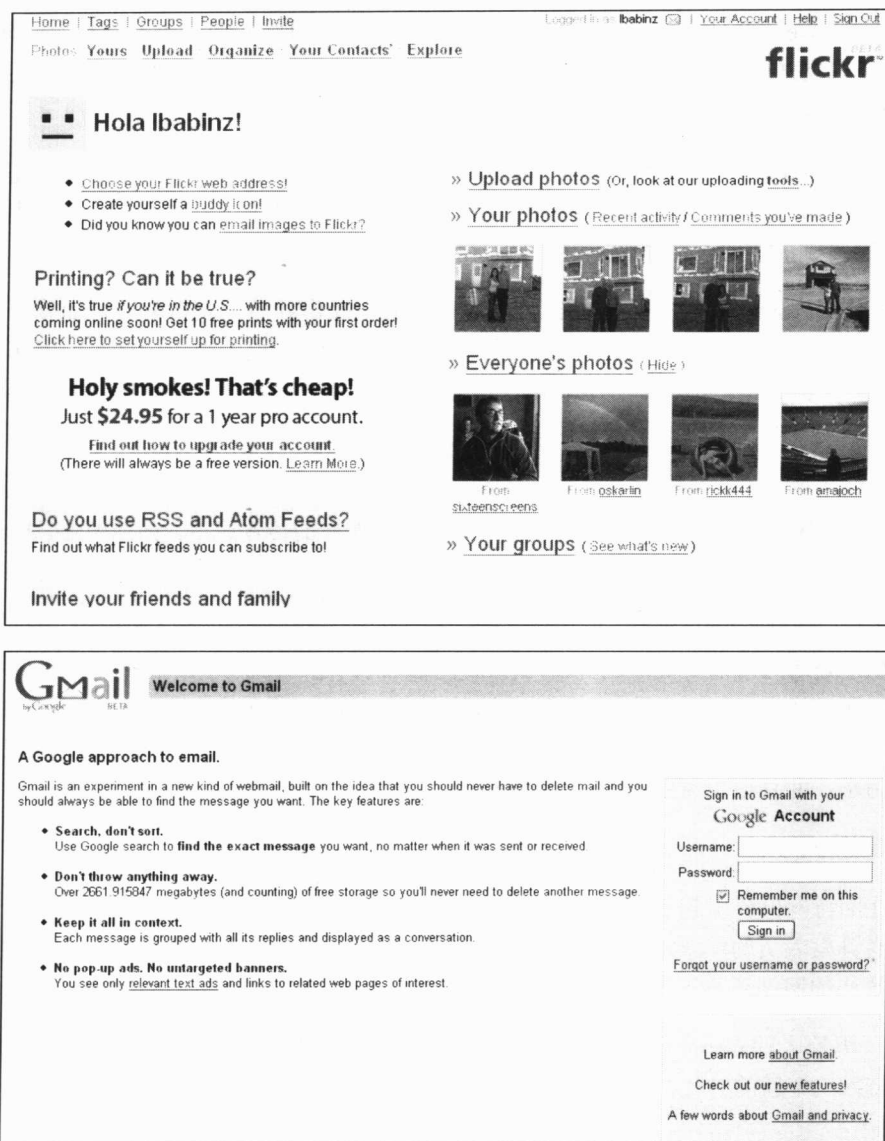


图1-2 诸如Flickr和Gmail之类的网站都是功能强大的Ajax应用

1. Ajax定义

Ajax表示“异步JavaScript和XML”，尽管不是所有人都认为Ajax是一个合适的术语，但那些

批评这个术语的人也不得不承认该技术之所以被广泛接受，这个新名词起到了一定的作用。

本质上，Ajax使用基于JavaScript的XMLHttpRequest对象，向Web服务器发送异步请求或者避免对页面进行刷新（图1-3和图1-4阐释了传统的和基于Ajax的请求/响应模型的区别）。通过使用XMLHttpRequest，Web应用程序能够向服务器获取或发送信息，由服务器完成所需的处理，然后动态地改变Web页面的某个部分，用户甚至不用转到其他页面或将焦点转移到其他地方。你可能认为通过XMLHttpRequest对象返回的数据格式都是XML，它当然能够返回XML，而且还能够返回在脚本语言中指定的任何数据格式。

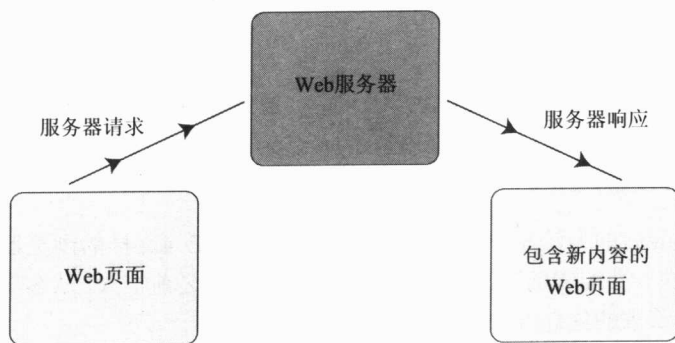


图1-3 现在绝大多数基于Web的应用程序都采用传统的服务器请求/响应模型：每次服务器请求发出后，该页面将刷新以显示新内容

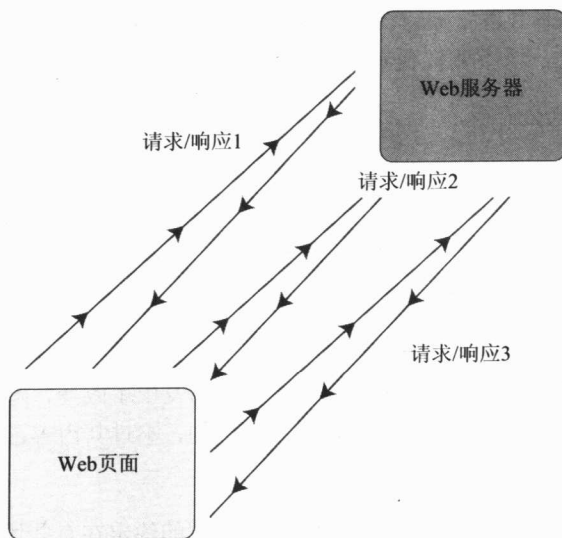


图1-4 使用Ajax异步方法的因特网请求/响应模型：页面可以发送多个服务器请求且无需刷新页面

看一个实例，使用贷款计算器表单来计算将会从银行账户中扣走的钱数，这对于脚本语言非

常简单。通常处理这种应用程序的方法是，填写表单内容，单击提交按钮，然后等待服务器响应。而且你可以重新再来一遍，以尝试新的财务公式。

然而如果使用了基于JavaScript的Ajax解决方案，用户单击提交按钮后仍然会停留在相同的页面中，服务器将完成计算并显示贷款值。然后可以直接修改表单中的值，并能马上看到不同的结果。

接着，修改功能具备了。甚至不用提交就可以通过Ajax在用户每次填写完字段后自动向服务器发送一个调用，并马上更新结果。像这样以人为本的友好功能已经成为了主流。

2. Ajax是新技术吗

在资深的Web开发人员面前说Ajax是新技术一定会遭到大声驳斥。Ajax不是新技术，事实上Ajax根本不算一项技术。Ajax描述的只是使用基于JavaScript的XMLHttpRequest对象以动态（异步）的形式从服务器获取信息的过程。

使用XMLHttpRequest的方法早在1998就开始流行了，如IE 4这样的浏览器就已经拥有实现Ajax的能力（尽管存在一些配置问题）。在当前浏览器开发出来之前很久，从客户端使用JavaScript来完成即时服务器端请求的处理方法就开始使用了。

不过如果从概念的角度（而非技术）来讨论Ajax的广泛使用，那么它对因特网社区绝对是一种新的启示。Web开发人员最终都会认识到，并不是所有向服务器发送的请求都要以相同的方式来处理。在某些方面，Ajax开放了被传统模式约束的上百万Web开发人员的思路，让他们看到冲破这些禁锢的可能。不过，请不要把我看作先锋，我只是这百万开发人员中的一员。

3. 现在Ajax为什么这么流行

该技术已经存在了这么久，为什么直到现在才开始流行呢？它是从哪里开始流行起来的，是谁推动了这个流行趋势，这都很难说清楚。也许是Gmail的应用广泛引发了许多开发人员的讨论，也许是Jesse James Garrett创造了这个术语使得这个概念有了统一的称呼；但我认为Ajax的真正意义，对于开发人员的要比对于使用者的更大。

我们来分析一下诸如会计这样的行业。多年来会计师都在使用纸制数据表及过时的数学公式去组织高度复杂的财务信息。随着计算机的出现，这一切发生了改变，出现了提供此类服务的新方法。的确，源于原始方法的标准直到现在仍然是正确的，不过其内容已经大大丰富了，而且运行业务的新方法也已经出现。

开发者们应用Ajax创建的因特网软件与此类似。传统的约定在有些场合仍然有效，不过现在有了一种实现功能和表现信息的新方法。它是一种新工具，可以应用并促进我们的业务。现在正在开发的一些新方法，直到近段时间才进入开发人员的视野。我对使用Ajax概念来构建应用程序感到十分兴奋，迫不及待想知道有什么创新的因特网机制可以使用。

1.2.2 Ajax的必要条件

因为Ajax是基于JavaScript技术的，因此用户的浏览器必须启用JavaScript才能够正常工作。也就是说，大多数用户可以选择浏览器是否使用JavaScript，这倒不会引起什么安全性问题。但必须注意，用户具有“禁用”Ajax的能力。因此当开发一个Ajax应用程序时，要确保网站中还有实现同样功能的其他方法，或者给用户一个提示告诉他运行该应用程序所需的环境，这点很重要。

Ajax已获得了许多浏览器的支持，其中包括Firefox（所有可用版本）、IE（4.0及更高版本）、Apple Safari（1.2及更高版本）、Konqueror、Netscape（7.1及更高版本）和Opera（7.6及更高版本）。因此绝大多数广泛应用的浏览器都提供了处理Ajax及其相应技术的手段。在第11章中将介绍更多与处理跨浏览器Ajax相关的问题。

要使Ajax的应用更加有效，唯一的要求就是打破多年应用的标准，然后创建一些真正革命性的、具有良好功能的东西。

1.3 小结

现在你应该对Ajax这个技术新宠的来源及其未来的发展方向有了比较清楚的认识。对于那些还没有动手试过Ajax的Web开发人员来说，读到这里时会很想知道它能够实现什么。我们首先引入了一个无需刷新页面的服务器请求执行方式的概念，只用了很短的篇幅阐述了所有现在可以实现的令人惊讶的构想。还讲述了该技术将打破所有传统约束。

做好了解更多知识的准备好了吗？让我们进入新的一章，学习如何在工作中应用Ajax和PHP。

对于Ajax有一个误解：能够实现这么酷的功能，JavaScript代码肯定是很难实现和维护的。刚开始尝试这一技术时的确并不简单。基于Ajax的服务器请求结构非常易于理解和调用。只需简单地创建一个XMLHttpRequest类型的对象，确定其是否创建成功、将放在哪里以及结果将在何处显示，然后发送它。这就是全部内容。

如果就只有这些，那么有什么可大惊小怪的？那是因为Ajax的关键不在代码，而在功能、友好性和用户界面方面。实际上从开发人员视角来看，实现Ajax其实并不难，它更多的属于锦上添花。这样，开发人员便无需过多操心代码是如何工作的，而是将精力集中在何时应用这个概念的构思上就行了。

虽然Ajax也能够用于很简单的目的，诸如载入HTML页面、执行表单验证之类的普通任务，但其威力只有在与强大的服务器端脚本语言协作时才发挥出来。本书中讨论的服务器端脚本语言是PHP。当把Ajax客户端交互概念与PHP服务器端引擎相结合时，就可以开发出令人惊叹的应用程序。这两者结合在一起的空间是有限的，而本书将展示如何通过它们的结合得到非常强大的功能。

要用Ajax和PHP来创建Web应用程序，首先必须对基础知识有深入的理解。注意，Ajax是一种JavaScript工具，因此当尝试理解Ajax类型的应用程序时，对JavaScript基础知识的掌握就尤为重要。让我们从基础知识开始。

2.1 HTTP 请求和响应基础

要深入理解Ajax概念的整体构思，那么理解网站是如何处理请求以及如何从Web服务器接收响应就很重要。现在浏览器从Web服务器获取信息所采用的标准是HTTP（当前版本是HTTP/1.1）。这是Web浏览器用来向网站发送请求，以及从Web服务器（负责返回响应）接收响应的方法。

HTTP请求的工作机制与电子邮件类似。也就是说，发送请求的同时传输一些报头（header），以便让Web服务器知道要提供什么服务以及如何处理该请求。虽然大部分报头都是可选的，但有一个是必需的（如果你要访问的不仅仅是服务器的默认页）——host。这个报头至关重要，用来告诉服务器要提供什么服务。

一旦服务器接收到一个请求，就决定返回什么响应。表2-1中列出了常用的响应代码。

表2-1 常用的HTTP响应代码

响应代码	描 述
200 OK	表示请求的文档或文件已找到，并正确返回
304 Not Modified	浏览器指出它拥有一个本地的缓存副本，而服务器端的内容与此相同
401 Unauthorized	请求所需文档需要验证
403 Forbidden	请求者对所请求的文档不具有相应的权限
404 Not Found	请求的文件找不到（例如不存在）
500 Internal Server Error	服务器在处理请求时遇到了问题
503 Service Unavailable	服务器因负载过大而无法处理该请求

另外，表单还存在多种请求方法。其中像GET和POST这样的方法，大家可能都很熟悉了。表2-2中列出了所有可能的请求方法（尽管通常只使用GET和POST方法）。

表2-2 HTTP请求方法

方 法	描 述
GET	发送请求的最常用手段；对服务器特定资源的简单请求
HEAD	类似于GET请求，只不过返回的响应中没有具体内容，用于获取报头
POST	用来发送包含用户提交数据的请求（对于基于Web的表单最合适）
PUT	传送当前请求文档的一个版本
DELETE	发送一个用来删除指定文档的请求
TRACE	发送请求的一个副本，以跟踪其处理进程
OPTIONS	返回所有可用的方法；可检查服务器支持哪些方法
CONNECT	用于SSL隧道的基于代理的请求

至此，你已经对如何从浏览器向服务器发送请求以及如何返回响应有了基本的认识，再来看看XMLHttpRequest对象的工作原理就简单多了。它实际上相当简单，只是在后台完成操作，无需刷新页面。

2.2 XMLHttpRequest 对象

Ajax只是一个概念，用来描述客户端XMLHttpRequest对象与服务器端脚本的交互。要通过Ajax向服务器发送请求，必须创建一个能够用于不同表单的对象。注意，在各种浏览器中XMLHttpRequest对象的实例化和处理方法是不同的。微软公司的IE是以ActiveX控件的形式创建该对象，而Firefox及Safari等浏览器则是使用基本的JavaScript对象。这对于希望在任何浏览器中都能执行Ajax功能的跨浏览器代码是至关重要的。

2.2.1 XMLHttpRequest 方法

XMLHttpRequest对象的实例创建之后，用户就能够使用它的一系列方法，如表2-3所示。根据使用对象的方式不同，重要的方法也不同。

表2-3 XMLHttpRequest对象的方法

方 法	描 述
abort()	取消当前的请求
getAllResponseHeaders()	以String类型的变量返回所有HTTP报头
getResponseHeader()	返回方法中指定的HTTP报头的值
open()	指定与服务器连接所需的各种属性，可以选择使用GET或POST等方法，确定连接模式是否为异步，以及指定要连接的URL
setRequestHeader()	发送请求时，在报头中添加一个标签/值对
send()	发送当前请求

虽然表2-3中列出的方法可能有些复杂，但并非如此，下面将更详细地分析。

1. abort()

abort方法非常简单，就是停止当前正发送的请求。通过该函数就能够很简单地控制连接的时间长度。如果希望明确地规定请求发送的时间，可以通过abort方法来提前结束请求。

2. getAllResponseHeaders()

可以通过该方法来获取传送的所有HTTP报头。以下就是一个报头集的示例：

```
Date: Sun, 13 Nov 2005 22:53:06 GMT
Server: Apache/2.0.53 (Win32) PHP/5.0.3
X-Powered-By: PHP/5.0.3
Content-Length: 527
Keep-Alive: timeout=15, max=98
Connection: Keep-Alive
Content-Type: text/html
```

3. getResponseHeader("headername")

可以通过该方法来获取特定报头的内容。这对于要查询部分报头信息时很有用，因为所有报头信息将是一个很大的字符串。例如，要查询所请求的文档大小，只需调用getResponseHeader("Content-Length")。

4. open ("method","URL","async","username","pswd")

现在才接触到XMLHttpRequest对象的最基本的部分。我们将使用该方法来创建与服务器上特定文件的连接。在这里可以设置打开一个文件的方法（GET还是POST），同时还将定义该文件将如何打开。注意，函数中每个参数并非都必须提供，可以根据实际情况来定。