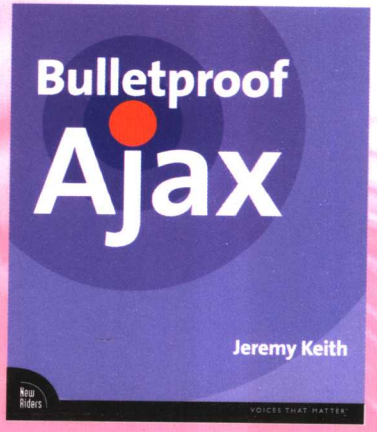
TURING 图灵程序设计丛书 Ajax系列

New Riders

Bulletproof Ajax 中文版

[英] Jeremy Keith 著
刘申 宋薇 译

- Web开发大师力作
- 剖析全新的Ajax开发方法——Hijax
- 深刻思想，令你醍醐灌顶



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 Ajax系列

Bulletproof Ajax中文版

[英] Jeremy Keith 著

刘申 宋薇 译

人民邮电出版社

北 京

图书在版编目（CIP）数据

Bulletproof Ajax 中文版 /（英）基思（Keith, J.）著；刘申，宋薇译．—北京：人民邮电出版社，2007.11
（图灵程序设计丛书）
ISBN 978-7-115-16626-5

Ⅰ．B… Ⅱ．①基…②刘…③宋… 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（2007）第 114911 号

内 容 提 要

本书介绍了如何构建无懈可击的 Ajax Web 应用程序，重点讲述如何在已有 Web 站点使用 Ajax 增强网站用户体验，从而尽可能地保证网站拥有最大限度的可移植性和亲和力，这正是目前大多数网站面临的需求。书中主要介绍了 JavaScript、DOM、XMLHttpRequest、数据格式等，同时还提出了一种 Hijax 方法，即可以让 Web 应用程序平稳退化的方法。

本书适合各层次 Web 开发和设计人员阅读。

图灵程序设计丛书

Bulletproof Ajax 中文版

◆ 著 [英] Jeremy Keith
译 刘 申 宋 薇
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：13.5
字数：235 千字 2007 年 11 月第 1 版
印数：1－5 000 册 2007 年 11 月北京第 1 次印刷

著作权合同登记号 图字：01-2007-1960 号

ISBN 978-7-115-16626-5/TP

定价：39.00 元

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

版 权 声 明

你的Ajax应用充满亲和力吗?

（译者序）

本书的作者Jeremy Keith是WaSP（Web Standards Project，http://webstandards.org/）的成员之一。WaSP的使命（http://www.webstandards.org/about/mission/zh-simplified/）中有一个主要的目标，即尽可能地保证网站拥有最大限度的可移植性（portability）和亲和力（accessibility），后者正是作者在本书中所要传达的核心内容之一。

对于大多数应用程序来说，无论是否基于Web，其操作都是基于文档的，比如Word、电子邮件，甚至是Photoshop。我们所处理的内容始终围绕着“文件”二字。Ajax的到来，使我们能够在不刷新网页的情况下处理其中的文档。这的确相当奇妙！

Ajax为我们带来新奇体验的同时，也带来了许多问题，其中最大的问题便是“缺乏亲和力”。为了解决这个难题，Jeremy Keith提出了一种方法，即Hijax（本书的第5章会对其作详细的介绍），这是一种可以让Web应用程序平稳退化（degrade gracefully）的方法。

亲和力只是Ajax面临的众多挑战之一，要想创建一个无懈可击的Ajax应用，需要考虑的事情还有很多。这些挑战，大多（我可没说是全部）不是技术上的，而是心理上的——是我们没有想到，或者根本不愿意去想的。我是WaSP国际联络组（WaSP International Liaison Group，http://www.webstandards.org/action/ilg/）的成员，在ILG的内部邮件列表里，可了解到各国在Web标准方面的进展与活动。如今，德国和瑞典都已经制定了各自的亲和力准则（accessibility guideline）。我真心希望，通过引进本书和其他传播Web标准理念的国外图书，亲和力的问题能够逐渐受到有关政府部门的重视，并建立起相应的准则，从而使中国的广大网民受益。

在此要感谢常可兄的引荐，使我有机会翻译本书。同时，也要感谢李琨和傅志红编辑的帮助与信任。

全书由刘申翻译，宋薇审校。由于水平有限，加之时间仓促，不妥之处在所难免，敬请广大读者批评指正。

刘　申

2007年5月于哈尔滨工业大学

前　言

如果你想了解什么是无懈可击的Ajax，本书就是你正确的选择。在这里，你会找到解决Ajax难题的方法，还有实际的例子演示。更重要的是，你会了解一些与Ajax相关的概念的解释和问题的答案。尽管书中有很多代码，但请不要仅把它看作一本教你如何编程的书，而应当把它看作一本指南，让它指引你在陌生的Ajax领域中遨游。

Ajax被认为是一种很奇妙的技术。我们所使用的大多数创建网站的技术基本可以分为两类：浏览器技术，例如HTML、CSS（层叠样式表）和JavaScript；服务器端技术，例如Apache、PHP和MySQL。而Ajax则介于浏览器与服务器之间。Ajax需要JavaScript——客户端脚本语言，但是它同样要与服务器进行通信。那么，它到底属于哪一类技术呢？

随着Ajax热度的逐渐升高，服务器端程序员被迫开始转向浏览器。他们具有多年软件设计和面向对象编程的经验，但是他们还没有准备好迎接浏览器端开发的挑战。与此同时，客户端开发人员也开始涉足Ajax，他们也将面临掌握这门新技术的巨大挑战。

许多书适合想学习Ajax的服务器端程序员阅读，但本书可能不是。如果你是一位Java程序员，并已习惯于创建复杂的对象，熟悉Web开发技术，包括JavaScript，也许应该更谨慎地选择本书。

如果你是一位前端开发者，那么本书正是适合你的。如果你已经精通Web标准，使用的也是语义化的标记和CSS，可能也了解一些基本的DOM编程，那么请继续读下去。

Ajax的学习之路可能有些令人迷茫，不过请不要担心，它并没有像宣传的那么难。正如你所看到的，JavaScript并不复杂，如何确保Ajax应用的无懈可击才是其中的难点。

2005年8月，New Riders出版了一本好书，Dan Cederholm编写的*Bulletproof Web Design*，书中的核心思想是灵活性。根据用户的不同需求，采用富有灵活性的设计元素，网站便可运行在不同的浏览器环境下。我认为这种思想同样适用于Ajax。

如今，许多Ajax应用都要依赖于浏览器中的某些特定技术，很不稳定，就像是用扑克牌搭建的房子。如果把不支持这些技术的浏览器通通拒之门外，会使得这些浏览器的用户因此而无法浏览。为了避免此种状况的发生，你需要使用一种无懈可击的Ajax方法来创建灵活的应用。

我为本书建了一个网站（http://bulletproofajax.com/），你可以去该网站下载并运行书中的实例（http://bulletproofajax.com/code/）。如果想了解最新的JavaScript和Ajax开发动向，请关注我的DOM Scripting博客http://domscripting.com/blog/。

致　谢

我从Dan Cederholm那里借用了bulletproof一词，并把它用在书名中。我欠他一个人情，外加一瓶美味的红酒。

本书的整个写作过程能如此顺利，要特别感谢Wendy Sharp的帮助。当初，是她引导我来写这本书。她的付出超出了我的想象：就在她打算举家远迁时，仍设法使本书得以最终成型。

感谢Jacqueline Aaron的文字编辑工作，是她对书稿进行了润色，使其更具文采。很怀念当初我们在一起讨论内容的表达方式、语法、标点的日日夜夜。

感谢我的朋友、同事兼技术编辑Aaron Gustafson。与Aaron共事，永远都是一件快乐的事。他不仅是一位JavaScript高手，还是一个很酷的人。

我也非常感激Joe Clark、James Edwards、Derek Featherstone、Bruce Lawson和Gez Lemon，感谢他们耐心地对第7章进行了审评。任何遗留的错误都只归咎于我个人的疏忽。

我在Clearleft的同事Andy Budd和Richard Rutter对我脱岗写作本书给予了极大的宽容。谢谢他们的理解。

我在2006年一系列的培训和讲座上对本书的许多素材进行了检验，感谢一路来听我唠叨这些内容的朋友们。得克萨斯州奥斯汀的South by Southwest、阿姆斯特丹的XTech、伦敦的@media和Barcamp、悉尼的Web Directions等活动是我思想的源泉，而且是很好的探讨问题的场所。谢谢Hugh Forrest、Edd Dumbill、Patrick Griffiths、Ian Forrester、John Allsopp、Maxine Sherrin和每一位背后推动这些杰出活动的人。

最后，我要特别感谢我的妻子Jessica Spengler，在我对写作产生倦意的时候，是她的支持与鼓励使我坚持下来。我爱你！

目　录

第1章

什么是Ajax

从古老的特洛伊到现代的Web设计

据荷马的《伊利亚特》记载，Ajax是Telamon（泰拉蒙）之子。身为一名希腊勇士，他以力量和勇气而闻名。在特洛伊之战中，他手拿大斧，身携巨盾。同时他还有一个很酷的名字。

Ajax这个名字太酷了，在《伊利亚特》中使用了不止一次。除了Telamon Ajax以外，Ajax the Lesser也是一名特洛伊战争中的英雄。从那以后，这个名字便不断被后人所用。

英国有一艘著名的战舰名为Ajax，它参加了第二次世界大战中的河床（River Plate）之战。漫画《飞侠哥顿》（Flash Gordon）里的那艘飞船也叫Ajax。而且至少有4款轿车、两张专辑、一支荷兰足球队和一款大型机的游戏曾经用过Ajax这个名字。当高露洁公司需要为它的家用清洁剂系列产品物色名字的时候，它们也选择了Ajax。

Ajax就是那种类似Excelsior（超越巅峰）或者Excalibur（亚瑟王的神剑）的词汇，很容易让人联想到一个充满力量的形象。或许是由于字母X的出现，外加其神话起源，这个词显现出何等的与众不同！

在网页设计这个充满时尚流行词汇的世界中，Ajax这个词不可避免地迟早会出现。

Jesse James Garrett 的故事

Jesse James Garrett是一位信息架构师、作家，还是位于旧金山的Adaptive Path公司的合伙创始人。2005年2月，他在Adaptive Path网站上发表了一篇名为*Ajax: A New Approach to Web Applications*（http://adaptivepath.com/publications/essays/archives/000385.php）的文章，如图1-1所示。

图1-1　Adaptive Path网站上有关Jesse James Garrett介绍

在这篇文章中，Garrett发明了Ajax这个词描述了一种新的Web应用所使用的技术。Google Suggest和Google Maps证明了基于浏览器的技术也可以提供那种在桌面应用中才有的交互和响应。但是，那时候还没有一个单独的词来表示这种已被应用的技术。

当这篇论文首次在Adaptive Path网站上刊登时，AJAX的4个字母是大写的。它原本是“Asynchronous JavaScript and XML”的缩写。这些单词的首字母恰好组成了我们所熟知的特洛伊勇士的名字，但是它却没有很好地对这种技术进行概括。

不错，大多数新一代的Web应用都是异步的（asynchronous）。也就是说，不用刷新浏览器，交互活动便可在后台进行。但是，正如我们稍后将会看到的，并不是非得使用异步传输不可。定义一个同步的交互是非常容易的。

代表XML的X更有问题，它似乎在暗示XML是Ajax应用中的一部分。这其实不对。实际上XML这几个字母也出现在`XMLHttpRequest`里（Ajax实现中的一项核心技术），但是，`XMLHttpRequest`听起来就不那么酷了。

Jesse James Garrett后来更新了这篇文章，澄清了Ajax并不是缩写。

尽管Ajax不是缩写，但是它却完美地概括了一组技术。它被编程大虾们嗤之以鼻。“这不是什么新技术，”他们嚷嚷说，“我们用它已经好多年了，称它为远程脚本（remote scripting）。Ajax只不过是个花哨的叫法。”

虽然像远程脚本这类技术词汇听起来没有像特洛伊英雄那么吸引人，但是在抱怨声中确实存在一个事实：Ajax所应用的这些技术中没有一项是新技术。不过，仍然没有理由就这样彻底摒弃这个词。

单词Ajax很简短，而且轻易地描述出一种涵盖多项技术的方法。它可使开发者和客户共同探讨当今Web应用中可用性和设计方面的相关问题。

但是，它的具体含义是什么呢？

1.1 诠释 Ajax

Jesse James Garrett新造的这个词在网页开发者中燃起了一把火。许多公司和个人都曾经对这种新方法进行过探索。现在终于有了一个新词可以用来描述他们的工作了。

在那篇文章发表了3个月后，Adaptive Path和O'Reilly在旧金山组织了一场Ajax研讨会。开发者和设计师汇聚一堂，展示各自的成果，讲述着Ajax如何改变了他们的工作方式。

会后，一位名叫Derek Powazek的与会者把Ajax描述为："如果说传统的网页是书信，那么Ajax便是即时通信。"（http://www.powazek.com/2005/05/000520.html）

在传统网站上，浏览器向服务器端请求整个页面。然后，用户点击一个链接或提交一个表单，此时需要浏览器向服务器发送一个新请求；之后服务器返回一个新页面，如图1-2所示。

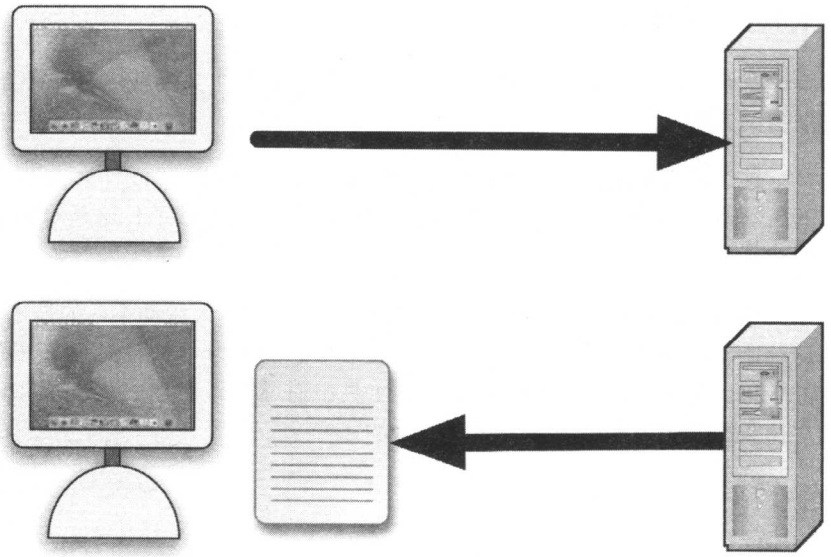


图1-2 Web的传统模型。客户端向服务器发送一个请求，服务器返回整个页面，如此反复

Ajax方法已经脱离了这种基于页面的模型。当用户与页面交互的时候（比如点击一个链接，提交一个表单，诸如此类），服务器只返回一部分信息。不用刷新整个页面，当前页面便得到了更新，如图1-3所示。

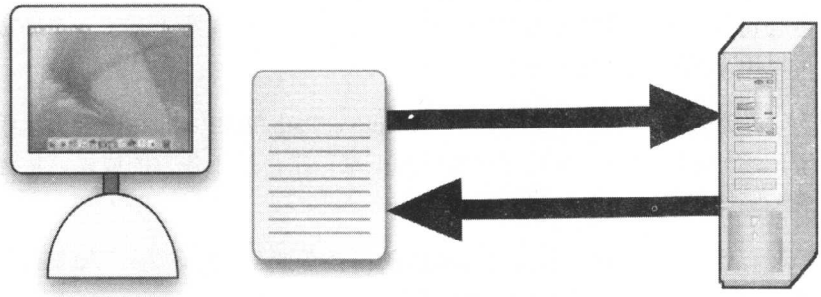


图1-3　在Ajax模型中，数据在客户端与服务器之间独立传输。服务器不再返回整个页面

这将给用户带来更流畅的体验。传统的网站带给用户的是开始-停止式的体验，Ajax应用则可以提供更快、更多响应的互动体验。

1.1.1　关键之处

我给了Ajax一个简单的定义：一种不用刷新整个页面便可与服务器通信的方法。

这个定义可能会产生两种反应。你或许会耸耸肩说，“没什么大不了的”；也许，你可能会瞪大了眼睛并惊呼，“真是太奇妙了！这是对Web的全新诠释！”

事实上，Ajax正处于两者之间。它确实是一项令人激动不已的技术，这种刷新网页某一部分便可从服务器获取信息的能力可以产生与众不同的效果；另一方面，Ajax仅仅是一种工具，通过它并不能产生更好的用户体验。还是内容为王。

1.1.2 种种选择

如果根据我的简单定义，许多技术都包含在Ajax概念之中。

1. Flash

Adobe Flash动画现在已可实现与Web服务器的异步通信了。这意味着不用刷新页面，就可以更新Flash动画内容了。听起来确实很像Ajax。

Adobe Flex框架为开发者提供了更强大的功能。如今的Flash已是一项成熟的技术，可用来开发功能强大的动态Web应用。（关于如何创建Flash应用，可以单独写一本书了。这还是交给别人去做吧。）

2. Java applet

Java applet是用Java编写的一些小程序，不要跟JavaScript弄混了。这些程序很像Flash动画，可以嵌在网页中。当它们被载入后，便能够与服务器进行交互。

applet的速度和响应根据不同的终端配置而有很大的差别。Java applet至今仍被广泛使用。

3. 框架

还记得框架（frame）吗？如今它们已经不是很常用了，主要是因为缺乏可用性。

如果你用一组框架构造了一个网页，可以只更新其中一个框架，而不必惊动整个页面。从技术上来说，根据我的定义，这就是Ajax。

你可别把我的话当真！我并不是说使用框架就可以去创建一个Ajax应用，但是它们确实有很多相似的地方。稍后我们将会看到，在Ajax应用中，许多由框架引起的可用性问题又重新出现了，比如收藏书签和浏览器后退按钮的未知行为等。

4. 隐藏的iframe

内联框架，即iframe，是由传统的框架组演变而来的。iframe可以作为连接Web服务器的秘密管道。如果网页中含有一个被隐藏了的小iframe，它的源代码就可以不断被更新。JavaScript能让父页面从更新过的iframe获取信息。

Google Maps就使用了一个隐藏的iframe来跟服务器进行通信。这是一个很巧妙的做法，尽管有点歪门邪道的意味。

5. XMLHttpRequest

XMLHttpRequest对象是对JavaScript的一个扩展，可使网页与服务器进行通信。它是创建Ajax应用的最佳选择。当Jesse James Garrett发明Ajax这个词的时候，他脑子里想的就是XMLHttpRequest。

XMLHttpRequest最大的问题就是名字太长了。即使名字中有一个字母X，要想成为一个流行词汇，它的易记性仍然是个问题。“Ajax”则显得更加简洁与帅气，通常把它当成XMLHttpRequest对象的代名词。它才是本书所要介绍的Ajax。

1.2　Ajax工具包

XMLHttpRequest对象是驱动Ajax的引擎，但是它并不是孤立存在的。正如Jesse James Garrett在其文章里所写到的，“Ajax并不是一项技术，它实际上是几种技术，每种技术各尽其职，以一种全新的方式聚合在一起。”

服务器端语言

若想让你的应用程序可以智能地响应浏览者的请求，那么服务器就需要具备向浏览器发送特定信息的能力。为了实现这一点，就要在服务器端使用某种编程语言。

服务器端的编程语言数不胜数：PHP、Java、Ruby、Python、Perl，等等。不能说哪种语言更适合Ajax。Ajax与服务器端的语言无关。你或你的团队可以使用任何一种自己擅长的编程语言。