TURING 图灵程序设计丛书 Web开发系列

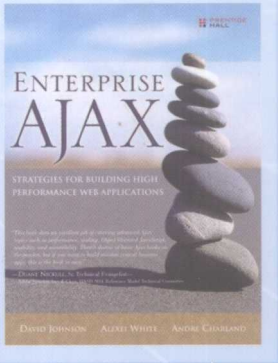
PRENTICE HALL

Enterprise AJAX
Strategies for Building High Performance Applications

AJAX企业级开发

David Johnson
[加] Alexei White 著
Andre Charland
张祖良 荣浩 高冰 译

- 第一部AJAX企业级开发力作
- 大量来之不易的专家建议和最佳实践
- 带你达到全新的高度



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 Web开发系列

Enterprise AJAX
Strategies for Building High Performance Applications

AJAX企业级开发

David Johnson
[加] Alexei White 著
Andre Charland

张祖良 荣浩 高冰 译

人民邮电出版社
北 京

图书在版编目（CIP）数据

AJAX企业级开发 /（加）约翰逊（Johnson，D.），（加）怀特（White.A.），（加）查兰（Charland，A.）著；张祖良，荣浩，高冰译 .—北京：人民邮电出版社，2008.10
（图灵程序设计丛书）
ISBN 978-7-115-18606-5

Ⅰ.A… Ⅱ.①约…②怀…③查…④张…⑤荣…⑥高… 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117088号

内 容 提 要

本书首先解释了AJAX为什么在大规模的开发中能有如此广阔的应用前景，接着系统地介绍了当前重要的AJAX技术和组件。你将看到把数据表、Web窗体、图表、搜索和过滤连接在一起用于构建AJAX应用程序的框架开发的整个过程；在此基础上，本书给出了已经过证实的AJAX架构模式，以及来源于实际的.NET和Java AJAX应用程序的案例研究。

本书适用于任何平台上的Web开发和设计人员。

图灵程序设计丛书

AJAX企业级开发

◆ 著 [加]David Johnson [加]Alexei White [加]Andre Charland
译 张祖良 荣浩 高冰
责任编辑 王慧敏

◆ 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
北京铭成印刷有限公司印刷

开本：800×1000 1/16
◆ 印张：18.75
字数：443千字 2008年10月第1版
印数：1－4 000册 2008年10月北京第1次印刷

著作权合同登记号 图字：01-2007-2673号

ISBN 978-7-115-18606-5/TP
定价：49.00元

读者服务热线：(010)88593802 印装质量热线：(010)67129223
反盗版热线：(010)67171154

版 权 声 明

前　言

也许你和我们所遇见的许多有才华的开发者一样，对AJAX技术以及如何使用这项技术来改善Web应用很感兴趣，你可能已经初步上网做了一番研究，访问过Ajaxian.com网站或者阅读了关于AJAX开发的入门图书。当然，你也可能属于人数更多的另一类有才华的开发者群体，想要走进AJAX世界，开始实际使用这项技术。无论是哪种情况，我们都做了考虑。令人高兴的是，开发者社区终于开始真正理解AJAX了。其实并没有那么难。

我们决定编写本书是因为我们对于现状很失望：关于AJAX开发更为高级的主题的信息太少了。主要原因可能是讲述这方面主题的图书仍然还在“编写”中，而且，尽管AJAX进入主流应用已有几年时间，但它才刚刚开始进入企业级软件开发的领地。我们希望本书能成为企业级开发者感兴趣的信息资源。为此，我们尝试把目前的开发方法与JavaScript以及其他组成AJAX的技术结合起来，并以所有企业级开发者都熟悉和易于理解的方式讲述。

为什么需要本书

本书大部分内容源自多年来我们在Nitobi公司（www.nitobi.com）构建AJAX应用和用户界面组件的第一手经验。这代表了我们在开发过程中的所知所得，对于那些希望把AJAX引入到开发项目中的开发者来说，这应该是很有用的资源。如果你想更加精通JavaScript开发，想解决AJAX怪癖[①]和性能问题，想从头设计可用性好的Web软件，那么本书将成为绝佳的资源。

我们有足够的时间来讨论如何以一种Java或者C#开发者熟悉的方式来编写JavaScript代码，并能使你快速上手。在这个过程中，我们会通过一些耳熟能详的软件设计模式来描述AJAX开发，并包含了AJAX开发过程中最热门的话题，例如安全性和离线存储。同时，不仅仅通过代码的优化，而且还通过利用因特网基础设施支柱（例如缓存），给出了构建高性能AJAX应用的真实解决方案。

本书采用了与其他AJAX图书略微不同的方法，讨论范围较为全面，其中包括关于编程方面的大量建议，以及应用可用性、可访问性和国际化等问题的丰富讨论。本书还包含了一个框架，用于AJAX开发项目中的风险评估。本书还特别介绍了在真实企业应用中使用AJAX的一些开发者，看看从他们的经验中能够学到些什么。

① quirks，直译怪癖，指早期的浏览器由于缺乏实践经验或者对Web理解存在问题，对于CSS和JavaScript的解释存在不一致性，各个浏览器都有自己的“怪癖”，容易出现代码的兼容性问题。——译者注

本书读者

本书主要面向中高级服务器端（Java、面向对象PHP或者ASP.NET）开发者。书中的很多概念来自Erich Gamma、Richard Helm、Ralph Johnson和John Vlissides所著《设计模式——可复用面向对象软件基础》一书提出的那些经过时间考验的软件工程模式。因为整本书都应用了这些思想，所以读者如果对软件设计模式有一个基本的理解，或者至少对学习这方面更多的知识有兴趣，会很有帮助。我们希望以一种常见的方式，即使用模式来描述AJAX，从而帮助更多有经验的开发者更容易地理解其中的概念和思想。

也许比理解模式更重要的是，你至少应该掌握JavaScript、HTML和CSS的基础知识，甚至是理解XML、XSLT或者JSON，这些知识也很有用，但不是必需的。除此之外，我们希望你拥有以面向对象的语言，如Java、C#或PHP等进行服务器端编程的经验。

阅读完本书之后，开发者应该熟悉组成AJAX的系列技术，以及面向对象JavaScript的开发。同时，你将很好地了解有助于开发应用的工具，以及诸如安全、可用性和可访问性等各种AJAX问题。

本书内容

第1章涵盖了AJAX应用的基本要素，并阐明了这些要素是如何组合在一起的。同时讨论了Web应用的演进，以及AJAX成为基于Web应用首选解决方案的主要原因。

第2章研究组成AJAX的各种技术。本章包含了关于使用正确的方法来编写JavaScript的重要信息，特别关注面向对象 JavaScript的开发、DOM、CSS、事件和`XMLHttprequest`对象，同时还包括了从客户端到服务器端数据传输相关的问题。

第3章是基于第2章内容的扩展，为理解主流浏览器的差异奠定了基础。具备了这些知识后，我们介绍了如何使用MVC（模型-视图-控制器，Model-View-Controller）模式来构建AJAX应用。特别是，你将明白如何在JavaScript中编写客户端模型，如何从数据生成 HTML视图，如何使用依赖发布-订阅（publish-subscribe）事件系统的基于JavaScipt的控制器来连接视图和模型。

第4章准备介绍如何构建用于Web应用的AJAX用户界面组件。特别地，我们分析了命令式和声明式方法的不同点，给出了一个构建基于AJAX的数据网格组件完整示例，同时还介绍了声明式方法的一些限制。

然后，本书给出了AJAX开发一些总体的目标和问题。第5章从应用设计到测试，再到部署，具体分析贯穿软件开发生命周期中AJAX特有的问题。阅读完本章之后，你将很好地把握各种AJAX性能问题以及对任意的AJAX开发项目自始至终都有用的许多工具。

第6章为读者介绍了AJAX开发过程中各种架构问题。其中包括异步消息通信模式的研究，以及与服务器通信的方法的研究，例如服务器推送（push）、缓存、负载和离线AJAX。虽然其中很多问题在任何基于Web的应用上都很常见，但这里我们将从一个独特的AJAX视角来讨论这些问题。

在第6章的基础之上，第7章讨论了AJAX如何在Web浏览器中使用Web服务来适应面向服务的架构，以及在构建AJAX Web应用时可能出现的各种安全问题。

第8章是本书的最后一部分内容的开始，讨论了可用性方面的一些问题，尤其是如何为普通

的用户把这些讨论应用到AJAX应用的构建中。本章介绍了人们所关注的常见问题的完整解决方案，例如后退按钮问题、处理可访问性以及国际化的方法。

第9章着手探索一些强大的AJAX用户界面模式，包括即时编辑（in-place editing）、主从复合结构（master-detail）、实时表单以及拖拽等。构建大多数的AJAX应用时，有许多核心的用户界面设计模式是每个开发者都应该了解的。

第10章把主题切换到了探索开发可伸缩的企业级AJAX应用时的风险来源。这个主题也许是AJAX书籍中探索最少的主题，但是当考虑构建新的应用时，它与技术本身同样重要。

第11章作为总结，分析了在要求最为严格的企业环境中的一些实际AJAX实现。我们与这些应用的开发者对话，并且倾听他们做对的和做错的事，以及下一次开发过程中将采取何种不同的实现方式。

总之，我们希望在AJAX开发方面给你一个新的认识，最重要的是，你可以把一些新的技巧引入到你的项目开发中。

支持和反馈

当然，我们会尽可能地保持本书所有信息的正确性和时效性，但是错误在所难免。我们预先为任何可能出现的错误致歉。请访问本书的网站以获取勘误表：http://www.enterpriseajax.com。

另外，你可以从本书的站点中方便地查找和下载到所有的源代码。获取源代码都需要有GPL许可。

我们同时也渴望得到关于本书、代码范例等内容的反馈信息，以用于下一个版本的改进。请直接将这些反馈提交到enterpriseajax@nitobi.com。

致谢

如果没有幕后这么多人的慷慨支持，就不可能有这本书稿。我们要感谢Prentice Hall出版公司，尤其感谢Mark Taub对整个过程的宏观把握。十分感谢Brent Ashley、Uche Ogbuji和John Peterson对本书非常有益的反馈。我们同时也要感谢在Nitobi的支持团队：James Douma、Jake Devine、Joel Gerard、Mike Han和Brian Leroux，他们在我们写作本书时接手了本来由我们负责的工作，同时还从技术上和编辑上进行了指导。

Dave Johnson：当然，我要感谢Alexei和Andre对于完成这个项目的帮助，同时还有幕后其他的一些人，例如Jordan Frank。当然我能保持头脑清醒，Kristin功不可没，我也将永远记住Jack的谆谆教诲。

Alexei White：除了已经提及的人，我还十分感谢我的合著者Dave和Andre，还有其他对这个项目作出贡献的人们，他们通过不同的形式给出了他们所有的专业的意见。他们是Bill Scott、Christian Van Eeden、Dave Huffman、Mike Hornby-Smith、Bob Regan、Gez Lemon和Luke Wroblewski。我还要感谢Lara，在我一心只想投掷飞盘的时候，鼓励我坐下来继续写书。

Andre Charland：首先，我要感谢我的合著者Dave Johnson和Alexei，感谢他们允许我与他们一起编写本书。这是一种荣誉和奖赏。我想要感谢我的母亲和父亲以及Jonny，在我想要退出时对我的鼓励。

目　录

第1章 AJAX和RIA

基于Web的传统应用在当今的企业中得到了广泛的应用，从客户关系管理（CRM）到企业资源计划（ERP）。尽管有用，但是这些应用中的大部分都在很大程度上依赖于传统Web应用的HTML表单，并且更多地通过服务器端编程来完成主要的工作。在这些传统的Web应用中，用户界面（UI）通常是死板的，无法与用户输入的数据进行交互，而需要完全刷新Web页面来把数据提交到服务器。组合一个新的基于HTML表单的界面，需要刷新整个Web页面，包括数据、样式和结构，以及所有的部分，这种方式将会因为长时间的延迟而导致十分糟糕的终端用户体验。

这正是AJAX的用武之地，它能够有效改善Web应用可用性。这种技术催生出了一种新型的Web应用，这种应用大大扩展了用户在Web浏览器里完成更多工作的可能性。AJAX不仅能够改善陈旧过时的Web架构，而且还可以使基于Web的应用在可用性和用户体验方面与桌面应用的地位相互竞争，甚至是超越桌面应用。AJAX甚至为强大的新型应用添加工作流和可视化功能，这些目前还没有基于桌面软件的同等价产品——倒不是因为桌面软件开发者缺少技术能力，而是AJAX已经使富因特网应用（RIA）为大多数Web开发者所触手可及。从这个角度来看，AJAX已经改变并且将持续改变用户对传统Web应用和桌面应用的看法。

虽然在最近一段时间内，AJAX已经从颇受欢迎的Google Web应用，诸如GMail和Google地图中获得了普遍的赞誉，但是事实上，AJAX技术和包括在AJAX首字母缩写词内的组成技术一起已经存在了将近十年。AJAX最初仅仅是动态HTML（DHTML）的重新命名，在过去，开发者社区尽量避免使用这项技术，如今却成为了一项热点技术。人们对大多数AJAX相关技术的理解都更加充分了。AJAX在公众型Web应用开发中非常流行，但这项技术同时在企业应用领域也开始取得了进展。本书把AJAX介绍给在企业中习惯于和传统Web应用打交道的开发者，包括了从CRM到电子商务应用开发各种类型。我们将介绍AJAX技术，提供一个坚实的技术细节基础，允许你构建高级的AJAX应用来改善应用的可用性，并且还将影响整个公司的业务。

我们猜你会问这样的问题："在企业领域，什么地方会用到类似AJAX这样的富客户端技术。"我们可以考虑AJAX带来的至少三个好处：

- AJAX能够为终端用户改善和增强用户体验，提高用户的工作效率和满意度。
- AJAX能够减少对网络和服务器基础设施的需求量，通过减少维护量甚至是减少带宽来节约费用，并且为所有的用户提高服务质量。
- AJAX能够开发在传统的应用模型中无法实现的新型功能，为用户完成目标提供新的工

具。

为了理解为什么存在以上这些好处，我们需要了解传统的Web应用模型所存在的无法跨越的局限性，以及AJAX如何从它的组成技术中实现更多的功能。改善Web体验的机会驱动了XMLHttpRequest、JavaScript和CSS的应用，并且为企业开发创造了新的机遇。

毫无疑问，企业级AJAX市场的机器正在高速运转。企业级供应商以多种形式支持AJAX。IBM创办了开放AJAX联盟（Open AJAX Alliance），且Dojo在Web开发讨论委员会中占据优势。微软发布了ASP.NET AJAX，Google发布了面向Java开发者的Web工具箱（GWT）。Oracle有ADF，一套用于JSF的AJAX组件。Yahoo发布了Yahoo用户界面库（YUI）。Adobe支持Flex和AJAX通过FA 桥（FA Bridge）交互，并且发布了名为Spry的开源AJAX框架。在这些项目的背后，是改善企业Web应用设计方式的真实而迫切的需求。

1.1　变化中的Web

在20世纪90年代末期，微软首次在IE 5中引入了XMLHttpRequest（XHR）对象，这个对象是AJAX功能所需的核心技术的一部分。同时，微软引进了Outlook Web Access（OWA），OWA是一个让人印象相当深刻的AJAX界面，而且在技术上远远超出它所处的时代。当时的主要缺点是无法在其他浏览器中使用XHR对象，并且对于锁定微软的又一个工具或者平台，社区存在强烈的抵触情绪。通过XHR在主流开发中直到现在才被缓慢采用，可以证实这一事实。

伴随着在Firefox和Safari对XHR远程脚本的最终引入，以跨浏览器的方式构建富异步通信才成为了可能。这也暗示着XHR能够被部署到更多不同用户的机器上，而不会引入太多风险。当XHR、JavaScript、DHTML和CSS结合时，创建富客户端应用且没有Web应用所特有的令人厌烦的刷新才成为了一种可能。与稍后介绍的其他很多富客户端技术不同，AJAX基于各种浏览器和操作系统都支持的开发标准，事实上消除了对厂商锁定的担忧，而且提高了可移植性。

在传统应用中，一切都是围绕Web页面是作为静态视图出现在应用中的这一做法的，而应用又是完全基于Web服务器的。用户唯一可能的交互是向Web表单输入数据或者是单击一个链接，这两种操作都导致了整个页面的刷新，而不管它是在CRM应用中更新一条完整的客户记录，还是在查看和编辑用户记录之间进行状态切换。在某些方面，传统的Web应用存在很多改进的空间——例如，当输入大量的数据时。同时，在很多情形下，传统的Web应用的确表现出色，例如搜索引擎或者文档储存库，长期以来都是传统Web应用成功的典范。此外，传统的Web能力，例如HTTP协议和资源缓存，对于基于AJAX的应用而言也非常有用。

不同于流行的AJAX地图和Email应用，大多数的企业级Web应用围绕数据录入、数据编辑或者数据报表构建。最常见的数据录入应用包括一个数据列表，例如CRM应用中的客户记录或者销售信息，数据条目能够添加、删除或者编辑。下面让我们分析这样的一个情况，在传统Web应用和基于AJAX的Web应用中，当一位出色的销售人员被要求使用一个慢得让人痛苦的新在线CRM工具在销售过程中跟踪会议记录、客户联系信息和销售进展信息时，用户的交互是如何进行的。

1.1.1 传统 Web 应用之痛

当推销员登录到应用时，他将面对一个包含了10个潜在客户记录列表的Web页面。对于大多数的传统Web应用，这类功能是通过使用静态的HTML<table>标签列出每条数据记录来实现的，列表附近是链接到编辑或者删除页面的按钮。销售人员现在想要基于一些新的信息更新记录。首要任务是找到需要更新的记录。如果在前10条记录中找不到，他将不得不进行搜索，通过翻页到下10条记录，在数据列表中导航查找，而且需要等待页面的刷新。找到这条记录后，用户单击编辑按钮。通过单击向服务器发送了一个请求。然后，一个包括许多表单字段的页面被发送给Web浏览器。大多数的表单字段是文本字段，其中有一些提供了复选框、下拉列表或者简单的数据校验（例如，检查本地电话号码确保其是7位数字）。在数据编辑表单中，除了传统的Tab和Shift+Tab功能键之外，不存在其他的键盘快捷键方式可在编辑字段中移动。在数据编辑完成之后，用户单击位于页面底部的保存按钮，把数据提交到服务器，从而服务器能够验证数据的正确性并且把数据提交到数据库。另外一个页面被发送回浏览器以确认保存操作。如果数据发生错误，用户在页面表单得到一个可视化的提示，这个页面需要被发送回浏览器，用户进行适当的编辑，然后再次单击提交按钮。如果每天执行很多类似的相同操作，这将是一种相当低效且乏味的过程。

我们宁可希望使用数据列表的Web页面作为编辑页面，这样每条记录都能够立刻被编辑，也不希望使用单独的表单编辑数据。在完成所有的修改后，我们能够同时将这些数据提交到服务器，然后进行保存。从可用性来说，这是很多传统Web应用所使用的用户界面类型，而并非使用上文所描述的单独的数据编辑方案。当用户保存数据时，所有的数据必须一次性保存，而不是在用户输入或者更新时增量地保存。这种方案意味着所有的数据必须被一次性大批量地发送到服务器，这将导致以下几个可能的结果：

- 并发或者校验问题迫使所有的数据以一种杂乱并且难以理解的方式重新显示，提示用户一次性修复数据的多个问题。
- 对于终端用户，没有任何的辅助手段重新提交数据时，断断续续的网络连接问题或者服务器错误可能会导致数据被破坏甚至是完全丢失。
- 用户认证失败，所有的改动将全部丢失。

无论结果如何，当服务器持久化数据到数据库并且重定向到新的Web页面时，通常会导致长时间重新刷新，从而导致终端用户极大的挫折感和痛苦。图1-1的时序图中展示了用户和系统之间的交互。尤其需要注意的是，用户闲坐在计算机前等待服务器响应的部分。（这个时间通常用来玩个人纸牌游戏。）

HTML表单对于某些类型数据确实有用，尤其是对于新手用户，或者没有频繁数据操作的界面。不过，对于那些必须进行快速导航和编辑的大量复杂的数据，则相当痛苦。如果用户需要从电子表格或者邮件中复制数据到应用，往往意味着重新录入或者复制并粘贴单独的数据片段。可用性专家有时把这种情况称为“转椅集成”（swivel chair integration），当然并不需要可用性专家来指出，这是一种低效的工作方式，一种糟糕的用户体验[①]。

① 转椅集成，指用户手工把数据输入到一个系统，然后把相同的数据输入到另外一个系统，这个短语源于用户使用转椅，从一个系统工作后转到另一个系统继续工作的情景。——译者注

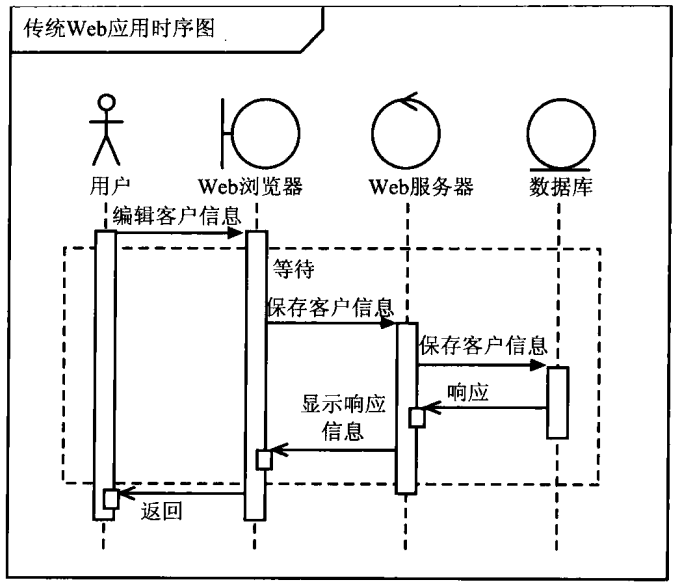


图1-1　传统Web应用数据编辑工作流的时序图（虚线框表示当服务器执行处理时，终端用户被迫等待的时间）

1.1.2　AJAX止痛药

与传统的Web表单处理大数据量的数据录入应用的方法不同，高效的应用需要具备响应性和直观性。总而言之，系统对于用户的工作流程的影响应该最小化。例如，用户需要在数以千计的预期的客户记录中上下滚动，犹如从本地计算机中访问数据，而不是每次翻页查看10条记录。同时，当数据被保存到服务器时，他们还需要继续输入数据到应用中。用户的使用习惯和与系统的交互方式必须尽可能地贴近桌面应用，从而减少用户把思维方式从桌面切换到Web上时所花费的时间。用于快速数据录入的理想界面需要类似于电子表格，但是每一列都要绑定到数据库表中特定的字段。尽管与传统的应用类似，同样使用简单的HTML<table>标签列出数据，但是当用户点击界面的任意数据时，该记录应该迅速变为可编辑状态，当用户按下回车键时，这些数据应该被保存到服务器，这种情况类似于大多数的电子表应用。如果在保存数据的过程中，由于数据库并发问题产生错误，当错误发生时，显示哪些数据出现错误的信息应该动态地显示在用户界面上。同样，在数据编辑完成并且按下回车键后，焦点应该自动地移动到下一条记录，而且这条记录在用户按下键盘上的任意键时立刻变为可编辑状态，正如我们所期望的桌面电子表格所能完成的一样。如图1-2所示，我们可以看到通过使用AJAX技术，用户不必再闲坐在计算机之前等待服务器的响应。相反，在保存操作的响应返回到浏览器之前，用户能够继续编辑数据。

基于AJAX技术的用户交互的关键在于，这项技术的核心是将少量的数据片段（而不是所呈现的HTML Web页面）发送到服务器以及从服务器读取，而不是发送由服务器完全装配的巨大的Web页面。这就是用户不需要等到数据保存之后才能发送请求到服务器，从而就能够继续编辑数据的原因。即使在这种情况下，由于在后台通过使用AJAX功能只把编辑过的数据异步发送到了

服务器，因此页面不需要刷新。

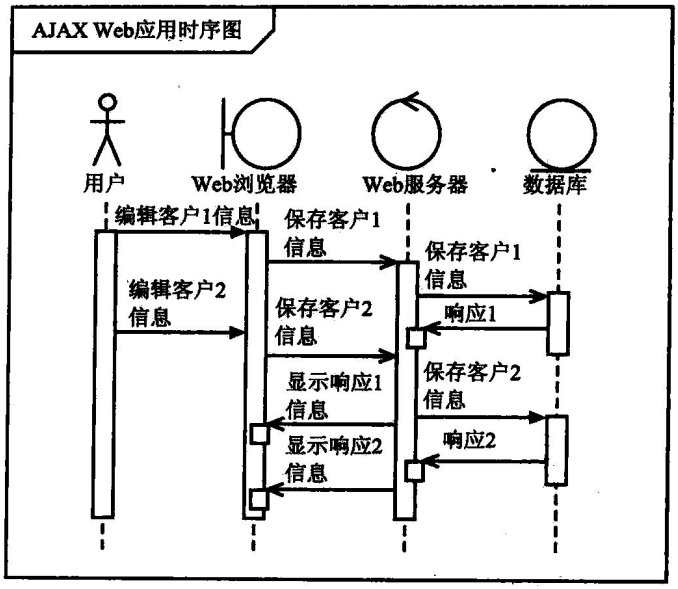


图1-2 AJAX Web应用数据编辑工作流的时序图（在服务器处理请求时，AJAX的异步特性允许终端用户继续工作）

在这个应用范例中，其他的热键同样可以工作，例如Ctrl+N创建一条新的记录，Ctrl+V从其他文本文档或者电子表格中直接粘贴到这个Web应用中（如图1-3所示）。此外，为了用户能够获取关于数据库中用户名和邮件地址是否可用的实时反馈，我们可以使用服务器端数据验证，从而进一步减少页面刷新的次数。

图1-3 演示桌面电子表格功能[例如基于鼠标激活特性的数据选择（例如数据的复制和粘贴）]的AJAX数据网格应用的屏幕截图

AJAX架构可用性的另一个优势是保护用户免受来自本身和网络的影响。花费了大量的时间填写一个很长的HTML表单，却仅仅因为失去了网络连接而无法将操作或录入的数据提交到服务器或者数据库，将是相当令人沮丧的。基于AJAX技术，我们通常能够把数据异步地发送回服务器。这项技术同时允许我们能够在任何时刻保持服务器端和客户端数据同步。尽管，我们不希望每次按键都不必要地提交对数据库的改动，但我们可以把数据推送到服务器，甚至是保存在本地，

从而避免由于网络停用或者客户端系统问题而丢失数据。

1.2　企业中的AJAX

直到最近一段时间，JavaScript才得到广泛的应用。由于各个浏览器之间不规范化的支持和安全性问题，JavaScript技术本身经历了一段在某些情况下被Web开发团队禁用的历史。在Firefox和IE中改进的JavaScript最终为开发者们提供了一个创建富应用的可靠平台，并且，AJAX术语的提出提供了一种通用的交流方式。一份BZ研究组织在2006年9月的调查报告（如图1-4所示）显示，18.9%的被调查者称他们的公司已经部署了使用AJAX技术的产品系统[①]。另外12.1%的被调查者称已经开发了他们的首个AJAX产品系统，但是还没有进行部署。同时，14.2%的被调查者正在开发实验性质的系统。另外，37.7%的被调查者称正在积极地研究这项技术。仅仅只有9.5%的被调查者称他们以及他们的公司都没有使用AJAX的计划（7.6%的调查者称他们不知道AJAX这项技术）。

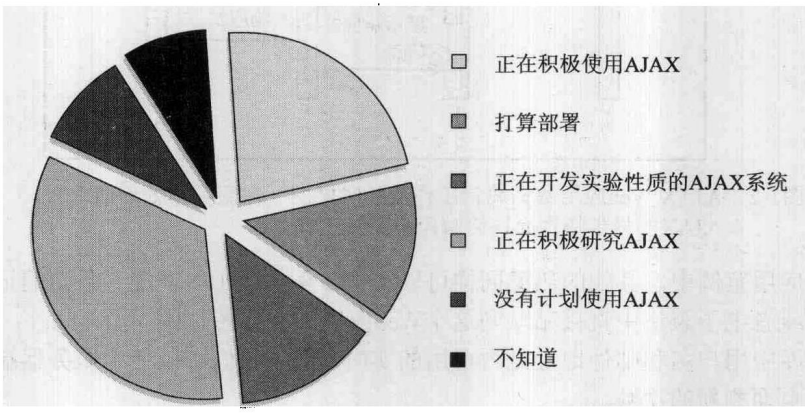


图1-4　2006年AJAX在企业中的应用情况（图片来源：SD Times）

当考察市场对合格开发者的需求时，与AJAX相关的新工作职位的数量令人吃惊。在图1-5中，我们可以看到要求AJAX技能的招聘职位的增长状况。

图1-5　AJAX职位趋势（图片来源：www.indeed.com）

① 参见http://www.sdtimes.com/article/story-20060901-12.html。