<<AJAX企业级开发>>

图书基本信息

书名：<<AJAX企业级开发>>

13位ISBN编号：9787115186065

10位ISBN编号：7115186065

出版时间：2008

出版时间：人民邮电出版社

作者：Davec Johnson,Alexeic White,Andrec Charland

页数：287

译者：张祖良,荣浩,高冰

<<AJAX企业级开发>>

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．tom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~~干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。
本书大部分内容源自多年来我们在Nitobi公司(www．nitobi．oom)构建AJAXJ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JavaScriptJ干发，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。

<<AJAX企业级开发>>

内容概要

《AJAX企业级开发》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，《AJAX企业级开发》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。
《AJAX企业级开发》适用于任何平台上的Web 开发和设计人员。

作者简介

本书的作者均来自著名的企业级用户体验组件和解决方案公司Nitobi。
Nitobi是OpenAjax联盟的活跃成员，具有丰富的企业级Web开发经验，多年来，为时代华纳、美国国家航空航天局、思科、宝马、宜家、朗讯、西门子等跨国公司和大型机构提供企业级解决方案，其中不乏关键任务和要求严苛的应用。

Dave Johnson Nitobi公司的CTO和创始人之一，也是业界知名的Web技术专家，长期从事高性能AJAX组件的架构和构建。

Alexei White Nitobi公司组件工具产品经理，是公司多个重要产品的主要架构师。

Andrec Charlandc Nitobi公司创始人之一，目前担任总裁和CEO，身经百战的Web技术专家，成功领导过100多个开发项目。

<<AJAX企业级开发>>

书籍目录

章节摘录

第1章 AJAX和RIA 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XIMLHttpRequest（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0utlookWebAccess（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。
　　1.1.2　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HTMLWeb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w曲应用中（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。

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HTML几表单，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，从而避免于网络停用或者客户端系统问题而丢失数据。

<<AJAX企业级开发>>

编辑推荐

《AJAX企业级开发》是第一部AJAX企业级开发力作，大量来之不易的专家建议和最佳实践，阐释企业应用的开发者经常遇到，但却很少能得到及时解决的问题，提供的准确信息和建议让你有信心在专业的业务应用中使用专业的Ajax技术，作者是著名Ajax技术公司Nitobi的创始人，该公司为OpenAjax联盟活跃成员。

版权说明