



Microsoft®
Press

Microsoft
核心技术丛书



Microsoft ASP.NET AJAX 导学

Introducing Microsoft ASP.NET AJAX

(意) Dino Esposito 著
成保栋 等译



机械工业出版社
China Machine Press

Microsoft
核心技术丛书

Microsoft ASP.NET AJAX 导学

Introducing Microsoft ASP.NET AJAX

(意) Dino Esposito 著

成保栋 等译



机械工业出版社
China Machine Press

本书详细地介绍 AJAX 的相关知识。主要包括: ASP.NET AJAX 技术概述、局部页面更新技术、Control Toolkit 方法、如何在客户端调用 Web 服务、如何创建客户端功能的组件等。本书内容全面、实例丰富,可以帮助读者快速掌握 AJAX 的相关技术。本书适合需要了解 AJAX 的技术人员参考。

Dino Esposito: Introducing Microsoft ASP.NET AJAX ISBN: 978-0-7356-2413-9

Copyright 2008 by Microsoft Corporation

Original English language edition copyright © 2007 by. Dino Esposito

All rights reserved. Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U. S. A.

本书中文简体字版由美国微软出版社授权机械工业出版社出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

版权登记号: 图字: 01-2008-2894

图书在版编目(CIP)数据

Microsoft ASP. NET AJAX 导学/(意)埃斯普斯托(Esposito, D.)著;成保栋等译. —北京:机械工业出版社,2008.6

(Microsoft 核心技术丛书)

书名原文: Introducing Microsoft ASP.NET AJAX

ISBN 978-7-111-24372-4

I. M… II. ①埃… ②成… III. ①主页制作—程序设计 ②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 086006 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 陈佳媛

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2008 年 6 月第 1 版第 1 次印刷

186mm × 240mm · 18.75 印张

标准书号: ISBN 978-7-111-24372-4

定价: 39.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线: (010)68326294

译者序

随着网站技术和 Web 2.0 概念越来越深入人心, AJAX (Asynchronous JavaScript And XML) 已广泛应用于各类网站和公司内部的应用程序平台。它的发展促使广大程序员, 由传统的服务器端 Render 开发方式逐步转变为由浏览器和应用程序平台的开发方式, 除此之外为改善系统性能, 提高系统运行效率的 AJAX 功能也在各种商业级应用程序中广泛使用。

但是, AJAX 需要针对浏览器客户端程序进行开发且采用标准的脚本语言格式, 而这类语言具有非编译性和难调试性的特点, 这就使得开发 AJAX 所需的成本和时间大量增加。

Microsoft 及其开发社区在 ASP.NET 2.0 平台之上发展了一套基于典型 AJAX 开发所需控件组, 这套控件集成了很多 AJAX 相关功能, 使得本需要大量客户端脚本的开发变得简单。本书就是主要介绍 ASP.NET AJAX 的。

什么是 AJAX

术语 AJAX 用来描述一组技术, 它使浏览器可以为用户提供更为自然的浏览体验。在 AJAX 之前, Web 站点强制用户进入提交、等待、重新显示的模式, 用户的动作总是与服务器的“思考时间”同步。AJAX 提供与服务器异步通信的能力, 使用户从请求、响应的循环中解脱出来。AJAX 可以在用户单击按钮时, 使用 JavaScript 和 DHTML 立即更新 UI, 并向服务器发出异步请求, 以执行更新或查询数据库。当请求返回时, 就可以使用 JavaScript 和 CSS 来更新相应的 UI, 而不是刷新整个 UI 页面。最重要的是, 用户甚至不知道浏览器正在与服务器通信: Web 站点看起来是即时响应的。

虽然 AJAX 所需的基础架构已经出现了一段时间, 但直到现在异步请求的真正威力才得以展现。拥有一个响应极其灵敏的 Web 站点确实激动人心, 因为它允许开发人员和设计人员使用标准的 HTML/CSS/JavaScript 创建桌面风格的 (desktop-like) 可用性。Google Maps 就是这类网站的典型代表, 它的成功告诉我们, Web 应用程序不必依赖于从服务器重新载入页面来向用户呈现更新。一切似乎都发生在瞬间。简而言之, 在涉及用户界面的响应灵敏度时, 基准设得更高了。

什么是 ASP.NET AJAX

如果说 AJAX 为 Web 增光添彩, 那么 ASP.NET AJAX 就是微软为 ASP.NET 披上的金履玉衣。那么什么是 ASP.NET AJAX 呢? 它涉及很多方面的内容。其中最为显著的优点就是它能够和 ASP.NET 实现无缝集成。ASP.NET AJAX 完全支持 Internet Explorer、Firefox 和 Safari 浏览器。

ASP.NET AJAX 是一个以快速、方便地搭建新一代的强大、互动、个性化、支持大多数

浏览器的 AJAX 网页为目标的免费框架。它集成了 Microsoft 浏览器客户端脚本资源和 ASP.NET 2.0 服务器端方便的环境。

ASP.NET AJAX 是 ASP.NET 的一个扩展,采用了 ASP.NET 服务器端的开发环境,因而 ASP.NET AJAX 将客户端脚本和强大的 ASP.Net 2.0 服务器端脚本相互融合。你可以更方便地创建绚丽、互动的 Web 应用程序界面。

采用 ASP.Net AJAX,你需要做的只是简单地拖拽几个控件到你的页面上,就可以使你的 Web 页面具有精彩的新一代 AJAX 用户界面的效果,同时大大地降低应用服务器层的资源消耗。

本书的主要内容和阅读对象

本书的作者 Dino Esposito 是 Microsoft.NET 技术方面的大师。本书既有原理性的讲解,也有示例说明。但本书并不是一本参考大全类型的图书,这位大师通过一些简单的例示,讲解了 ASP.NET AJAX 实现的技术原理、基本原则、良好习惯、正确思路和应该注意的设计模式。能从本书中领会到这位大师的思想精髓和实践经验,就是读者的最大收获。如果读者能够通过书中的几个案例真正领悟这些 AJAX 开发的原则、习惯和思路,就一定能让自己的 Web 编程技术再上一个台阶。

在此感谢我的朋友王立勇、朱东峰、周瑞涛、张昱等对本书翻译和校对工作的积极参与,也感谢我的家人对我翻译工作的大力支持,没有他们的帮助,就无法早日完成本书的翻译。

您是否已经迫不及待地要一览究竟了。那么别再犹豫,让我们一起跟随 Dino Esposito 大师进入 ASP.NET AJAX 的世界吧。最后,祝您阅读愉快。

成保栋

2008 年 3 月 22 日

作者简介

Dino Esposito 是意大利首都罗马的一位软件培训师兼软件顾问。他是 Solid Quality Mentors (www.solidq.com) 团队的成员之一, 专攻 Microsoft.NET 技术, 他时常在欧洲、澳大利亚及美国从事技术和咨询工作。

多年来, 他在为银行和保险公司搭建体系结构和创建分布式系统方面, 积累了第一手的经验和技能, 这些经验和技能也同样适用于对安全、优化、性能、可伸缩性和交互性要求极高的应用背景。

全世界每个月至少有 5 家期刊和网站会发表 Dino 的文章, 这些文章的主题包括从 Web 开发到数据存取, 从软件最佳实践到 Web 服务等。作为一名多产的作家, Dino 每个月都会为 MSDN Magazine (<http://msdn.microsoft.com/msdnmag>) 的 “CuttingEdge” 专栏, ASP.NETPRO Magazine (www.ASPNETpro.com) 的 “CoreCoder” 专栏, 以及 DrDobb's Journal (www.ddj.com/dept/windows) 的 ASP.NET-2-The-Max 时事通讯专栏撰写文章。作为一位在使用 NET 技术创建 Web 应用程序方面最具有权威和公众认可的专家, Dino 为微软的开发者和 IT 技术的推广做出了巨大贡献, 在 MSDN Developer Centers, 如 ASP.NET (msdn.microsoft.com/ASP.NET), Microsoft Windows Vista, 以及数据访问专栏, 都可以看到他的大量文章。

Dino 已经出版了一系列书籍, 其中大部分书籍都被认为是该领域内经典级的著作, 其中包括《Applied XML Programming with the .NET Framework》于 2002 年由微软公司出版。他创作的《Visual C++ Windows Shell Programming》(Apress, 1998) 至今仍然被全世界的开发人员广泛使用、讨论和引用。对于需要扩展 Windows 操作系统功能的开发人员, 该书仍具有其独特的参考价值。

他最新的两本著作是《Programming Microsoft ASP.NET 2.0: Core Reference》和《Programming Microsoft ASP.NET 2.0 Applications: Advanced Topics》(于 2006 年由微软公司出版)。作为 ASP.NET 方面微软最有价值的专家 (MVP), Dino 定期地在国际业界技术会议 (Microsoft TechEd、Microsoft Dev Days、Dev Connections、DevWeek 和 Basta), 以及欧洲和美国当地的技术会议上发表演说。

ASP.NET 发展至今, 已经从一个简单的 Web 开发工具, 成长为一个强大的 Web 开发平台。ASP.NET 2.0 的发布, 更是为 Web 开发人员提供了一个全新的开发环境。在 ASP.NET 2.0 中, 开发人员可以利用其强大的功能, 轻松地开发出高性能、可扩展的 Web 应用程序。ASP.NET 2.0 的发布, 标志着 ASP.NET 技术进入了一个新的阶段。开发人员可以利用其强大的功能, 轻松地开发出高性能、可扩展的 Web 应用程序。ASP.NET 2.0 的发布, 标志着 ASP.NET 技术进入了一个新的阶段。

前言

AJAX 表示“**Asynchronous JavaScript and XML**”，它是 2005 年提出的一个术语，指用于创建丰富的、高度交互性和响应性的 Web 应用程序的众多技术，主要是应用程序利用客户端进行大量工作，并向服务器进行带外调用(out-of-band call)。带外调用是一种服务器请求，它将会导致页面的更新，而不是页面替换。这一技术的效果让 AJAX Web 应用程序在一定程度上更像是传统的桌面 Microsoft Windows 应用程序，具有桌面程序的许多高级功能，如拖放、异步任务、高度的程序响应和没有抖动的用户界面，并保证用户功能的可用性。

ASP.NET AJAX Extensions 是对 ASP.NET 平台的一个重要扩展，可以用更加合理和有效的方式来实现 AJAX 风格的功能。因为已经将 ASP.NET AJAX Extensions 设计为 ASP.NET 的一部分，所以它可以同现有的平台和应用程序模型无缝集成。

从体系结构上来说，ASP.NET AJAX 框架由两个完全不同的元素组成：一个客户端数据库和一组服务器端扩展组件。客户端脚本库完全是用 JavaScript 编写的，因此，可以在现有的任何浏览器中运行。服务器扩展组件完全与以 ASP.NET 服务器为基础的各种服务和控件相集成，这样，开发人员就可以用他们所熟悉的、与开发传统的基于 ASP.NET 服务器页面类似的方式来创建丰富的 Web 页面。

大多数 ASP.NET AJAX 开发人员以前都从事过 ASP.NET 开发，所以他们对以控件为基础的服务器端开发模型会相当熟悉。以服务器为中心的编程模型是 ASP.NET 编程模型发展史上迈出的重要一步。ASP.NET AJAX 服务器端控件的功能非常强大，尤其是开发人员对手工创建 AJAX 客户端脚本不太熟悉时更为有用。

本书通过大量实例来讲述 ASP.NET AJAX 的大致框架，让读者可以熟悉种种相关的技术和工具。

对于 ASP.NET 开发人员和专业的人士来说，AJAX 真是一个重要的突破。它使得跨浏览器编程成为现实，让 Web 程序具有了类似桌面程序的功能。

谁应该阅读这本书

本书适合任何 ASP.NET 开发人员和专业人士阅读。如前所述，ASP.NET AJAX 是 ASP.NET 发展当中的一个重要的里程碑，它反应了业界的一种基本倾向——AJAX 模型。此外，ASP.NET AJAX 超越了经典的 AJAX 模型，它的框架可以跨越客户端和服务端，为 Web 应用程序提供点对点的解决方案。就 Microsoft Web 平台来说，ASP.NET AJAX Extensions 包罗了各种各样的功能(Web 专业人员的夙愿最终得以实现了)。

如果你是一名 Web 专业人员，正在使用基于 Microsoft 的 Web 技术进行开发，AJAX 一定是你值得抓住的一个机遇。这本书就是你的起点！甚至会收获更多。

本书是如何组织的

本书共分为三个主要部分：对平台和它的构建模块的综述，用于有效地增强现有网站性的技术，和以客户端为中心的开发。在第一部分中，将讲述有关 AJAX 模型的基本概念，以及为了支持这一模型而对 JavaScript 语言做出的扩展。第二部分主要介绍框架中用于扩展现有的服务器端控件的构建元素，以及如何将现有的传统 ASP.NET 页面转换为功能完善的 AJAX 页面。最后在第三部分中介绍了众多可以展示 AJAX 应用程序强大功能的开发工具和技术(对服务器端代码的带外调用)。

系统要求

为了可以运行本书的代码例子，需要具有以下硬件和软件：

- 支持的操作系统：Microsoft Windows Vista、Microsoft WindowsXP (安装了 ServicePack2)、Microsoft Windows Server2003 (安装了 Service Pack1) 或 Microsoft Windows 2000 (安装了 Service Pack 4)
- Microsoft Visual Studio 2005 Standard Edition 或 Microsoft Visual Studio 2005 Professional Edition
- Microsoft SQL Server 2005 Express (已经包含在 Visual Studio 2005 中) 或 Microsoft SQL Server 2005
- 具有 Intel PentiumIII 600MHz (或同等性能的处理器的) 或更高处理器 (建议使用 1GHz 或更高的处理器) 的计算机
- 最低 192MB 的 RAM (建议使用 256MB 或更高的 RAM)
- 至少支持 256 色，800 × 600 或更高的分辨率的显示器 (推荐用 1024 × 768，16 位真彩色)
- Microsoft 鼠标或其他兼容的定位设备

配置 SQL Server 2005 ExpressEdition

本书中一些章节的例子需要可以访问 SQL Server 2005 Express Edition (或 SQL Server2005) 来创建和使用 Northwind Traders 数据库。下面以 SQL Server 2005 Express Edition 为例，介绍为用户账号进行授权的步骤，使用经过授权的账号才可以运行本书中的示例代码。

1. 使用具有管理员权限的用户账号来登录机器上的 Windows 系统。
2. 在 Windows 的“Start”(开始)菜单中，点击“All Programs”(所有程序)，点击“Accessories”(附件)，再点击“Command Prompt”(命令提示符)来打开命令提示符窗口。
3. 在命令提示符窗口中，输入以下区分大小写的命令。

```
sqlcmd -S YourServer\SQLExpress -E
```

将 *YourServer* 替换为你的电脑主机名。

在运行 `sqlcmd` 命令之前，可以先在命令提示符窗口中运行 `hostname` 命令来查询计算机的主机名。

4. 在 1 > 命令提示符后，输入以下命令，包括方括号，再按回车键：

```
sp_grantlogin [YourServer\UserName]
```

将 `YourServer` 替换为相应的电脑主机名，将 `UserName` 替换为你想要使用的用户账号。

5. 在 2 > 提示符后，输入以下命令，并按回车键来确认：

```
go
```

如果出现错误提示信息，请先确认是否正确输入了 `sp_grantlogin` 命令，包括方括号。

6. 在 1 > 命令提示符后，输入以下命令，包括方括号，再按回车键：

```
sp_addsrvrolemember [YourServer\UserName], dbcreator
```

7. 在 2 > 提示符后，输入以下命令，并按回车键来确认：

```
go
```

如果出现错误提示信息，请先确认是否正确输入了 `sp_addsrvrolemember` 命令，包括方括号。

8. 在 1 > 提示符后，输入以下命令，并按回车键来确认：

```
Exit
```

9. 关闭命令行提示窗口。

10. 注销管理员账号。

Northwind Traders 数据库不再随 SQL Server 2005 一同发布，需要单独下载。可以从以下网站下载 Northwind 数据库的安装脚本：<http://www.microsoft.com/downloads/details.aspx?FamilyId=&displaylang=en>，在下载页面中有详细的安装说明。

代码示例

书中所有的代码示例可以从本书的网址 <http://www.microsoft.com/mspress/companion/9780735624139/> 下载，要下载示例文件，请单击该网页中的“Companion Content”（附带内容）链接。下载的代码包括了大多数章节的工程文件、代码片断和引用的例子。

本书支持

我们在编辑时尽了最大的努力，以确保本书和附带内容准确无误。所搜集到的错误反馈和修订将会及时在 Microsoft 知识库 (Microsoft Knowledge Base) 的文章中公布。

微软公司在以下 Web 站点为其发行的图书和附带内容提供支持：<http://www.microsoft.com/learning/support/books/>。

问题和评论

如果你对本书及其配套内容有什么评论、问题或想法，或有通过访问上述网站无法找到

答案的问题，请发电子邮件给 Microsoft Press，邮箱地址是：

mspinput@microsoft.com

或发普通邮件到：

Microsoft Press

Attn: Introducing Microsoft ASP.NET AJAX Editor

One Microsoft Way

Redmond, WA 98052-6399

请注意，以上地址不提供 Microsoft 软件产品的技术支持。

致谢

AJAX 是 Web 开发人员最应该关注的技术之一。本书对任何 Web 开发者都适用，不论 Web 开发人员的技术背景如何，本书都为他们学习 AJAX 提供了平稳而有效的开端，编著一种新技术的图书总会具有一定的难度，因为作者不仅要自学新知识，还要不断寻找如何向别人解释和传授这种技术的最好方式。

我也通过这本书来学习 ASP.NET AJAX，所以我将是这本书的第一个受益人。在我写完这本书之后，可以说，我相信自己对这一技术了解得已经相当多了。因为我学习能力强，还是因为这本书是本好书？对此我也不好下结论。

非常感谢 Scott Guthrie 花时间与我交流，向我解释技术发展方向和在 2005 年的 Professional Developers Conference 会议前后开发团队的学习情况，正是在那次会议上第一次公布了“Atlas”平台。Simon Calvert 为这一平台提供了个人指导，Stefan Schackow 为一些疑难问题提供了富有洞察力的答案。我贪婪地阅读了 Bertrand LeRoy 和 Nikhil Kothari 的大量日志（这些日志讲述的主题包罗万象），非常不错，我发现它们非常有用。

在编写这本书的过程中，我也联系了一些在 AJAX 领域活动的经销商，感谢来自 ComponentOne 的 Miljan Braticovic、Infragistics 的 Tony Lombardo 还有 Telerik 公司的 Svetozar Georgiev。ASP.NET AJAX 确实非常伟大，但这些朋友和他们的团队工作得也很出色，他们为应用程序创建的组件可以让 Web 程序像桌面程序一样运行。

Christian Gross 刚写了一本关于 REST 开发的著作，有助于我理解 REST 的概念以及它在 AJAX 和 ASP.NET AJAX 中所扮演的角色（在第 8 章有相关介绍）。

有关这本书的创作过程，说来话长，可以这么说，在编写本书时，我周游了 3 个大陆和 9 个国家：意大利、美国、法国、西班牙、奥地利、澳大利亚、德国、英国和马来西亚。我的朋友 Greg Linwood 和 Brett Clarke 向我挑战，看我能不能在一周之内完成一章的创作，同时还得做培训，到墨尔本观光，并为一个非常重要的纪念日准备一场特别的 Microsoft PowerPoint 介绍。我真把这些活一气呵成，相当有趣！

Fritz Lechnitz 也向我挑战，看我能不能在参加维也纳一场大型网球锦标赛的同时，也完成一章的创作。在那场比赛中，在对方顶尖选手暴风雨般的反手和正手球的攻击下，我被打得落花流水、晕头转向。

Andrea Saltarello 也向我挑战, 在 ASP.NET AJAX 官方正式版本发布一周之后, 就请我上一堂传授 ASP.NET AJAX 的课。我们把它外包了出去。

最后, SolidQ 的朋友也等不及我完成这本书, 就开始发起一些大型的培训了。

还有谁? 哦, 当然, 还有可爱的 Lynn Finnel 和整个编辑团队, 他们非常灵活、开放, 允许我把交稿日期一推再推。Kenn Scribner 发现了 ASP.NET AJAX 和我, 并不止一次地将我引向正确的轨道。(当然, 我也会回报她的好意, 但这经常是出于友谊……)

Ben Ryan 非常想看到这本书, 现在这本书终于面市了。感谢 Ben!

一如既往, 感谢我的妻子 Silvia, 还有我的孩子 Francesco 和 Michela, 我每写完一本书他们都会长高许多。

目 录

译者序

作者简介

前言

第一部分 ASP.NET AJAX 构建模块

第 1 章 AJAX 的发展	2
1.1 变革的过程	3
1.1.1 传统的 Web 应用程序	4
1.1.2 基于 AJAX 的 Web 应用程序	5
1.1.3 Rich 浏览器的角色	7
1.2 AJAX 核心引擎	8
1.2.1 XMLHttpRequest 对象	9
1.2.2 HTTP 对象模型	12
1.2.3 运行自制的 AJAX 框架	14
1.2.4 再谈文档对象模型	19
1.2.5 ASP.NET 中现有的 AJAX 框架	22
1.3 ASP.NET AJAX 实践	29
1.3.1 搭建 ASP.NET AJAX Extensions 环境	29
1.3.2 核心组件	30
1.4 小结	32
第 2 章 Microsoft 为 AJAX 提供的客户端库	33
2.1 JavaScript 语言扩展	33
2.1.1 扩展机制的基础结构	34
2.1.2 面向对象的扩展	44
2.2 核心组件	49
2.2.1 Sys. Application 对象	50
2.2.2 Sys. Component 对象	52
2.2.3 网络协议组件	55

2.2.4 用户界面组件	57
2.2.5 其他组件和功能	61
2.3 小结	63

第二部分 为网站增加 AJAX 功能

第 3 章 ASP.NET AJAX 的核心	66
3.1 ASP.NET AJAX 的配置	67
3.1.1 Web. config 文件	67
3.1.2 运行时引擎	69
3.2 脚本管理器组件	71
3.2.1 ASP.NET ScriptManager 控件	71
3.2.2 加载脚本	75
3.2.3 脚本错误处理	81
3.3 小结	85
第 4 章 局部页面更新	86
4.1 定义局部可更新区域	86
4.1.1 UpdatePanel 控件概述	87
4.1.2 启用页面局部更新	90
4.1.3 检测正在执行中的更新操作	95
4.1.4 UpdatePanel 控件的运行机制	104
4.2 控制可更新区域	109
4.2.1 触发面板更新	109
4.2.2 触发周期性的局部更新	114
4.2.3 在局部更新过程中提供用户反馈	116
4.2.4 局部更新的客户端事件	120
4.2.5 在局部页面更新期间传递数据项目	126
4.2.6 在局部更新期间让面板运用动画效果	131
4.3 小结	134
第 5 章 AJAX Control Toolkit	135
5.1 扩展器控件	135

5.1.1 扩展器究竟是什么	136
5.1.2 ExtenderControl 类	138
5.1.3 扩展器创建示例	140
5.2 AJAX Control Toolkit 简介	146
5.2.1 安装 Toolkit	147
5.2.2 AJAX Control Toolkit 中的 各控件	150
5.3 Accordion 控件	152
5.3.1 Accordion 控件概述	152
5.3.2 使用 Accordion 控件	154
5.4 Rating 控件	155
5.4.1 Rating 控件概述	156
5.4.2 使用 Rating 控件	158
5.5 ReorderList 控件	159
5.5.1 ReorderList 控件概述	159
5.5.2 使用 ReorderList 控件	161
5.6 TabContainer 控件	163
5.6.1 TabContainer 控件概述	164
5.6.2 使用 TabContainer 控件	165
5.7 AJAX Control Toolkit 的扩展器	166
5.7.1 面板扩展器	166
5.7.2 Button 扩展器	170
5.7.3 弹出式扩展器	173
5.7.4 用户界面扩展器	178
5.7.5 输入型扩展器	186
5.7.6 Animation 扩展器	199
5.8 小结	201

第三部分 以客户端为中心的开发

第6章 内建的应用程序服务	204
6.1 表单认证服务	205
6.1.1 身份认证服务的系统基础 架构	205
6.1.2 在应用程序中使用身份认证	

服务	209
6.2 用户个性化服务	217
6.2.1 用户个性化服务的系统 基础架构	218
6.2.2 在应用程序中使用用户个性化 服务	221
6.3 小结	226
第7章 ASP.NET AJAX 远程方法 调用	227
7.1 为远程调用设计的服务器 API	228
7.1.1 定义远程 API 的契约	229
7.1.2 实现远程 API 的契约	230
7.2 通过 Web 服务进行远程调用	232
7.2.1 创建 AJAX Web 服务	233
7.2.2 使用 AJAX Web 服务	236
7.2.3 AJAX Web 服务设计原则	242
7.3 通过页面方法实现远程调用	245
7.3.1 创建页面方法	245
7.3.2 使用页面方法	247
7.4 桥接外部的 Web 服务	251
7.4.1 传统的服务器对服务器方法	251
7.4.2 ASP.NET AJAX Futures 的 桥接文件	252
7.5 小结	260
第8章 用 ASP.NET 创建 AJAX 应用 程序实例	261
8.1 AJAX 深入解析	262
8.1.1 AJAX 的优点	262
8.1.2 AJAX 的缺点	264
8.1.3 模式、实践和服务	266
8.2 重访 ASP.NET Starter Kits	270
8.2.1 Jobs Site Starter Kit 速览	271
8.2.2 减少页面抖动	272
8.2.3 定期刷新屏幕	278
8.3 小结	287

ASP.NET AJAX 构建模块

第2章 Microsoft 为 AJAX 提供的客户端库

第2章 Microsoft 为 AJAX 提供的客户端库

现在的 Web 应用程序从体系结构和实现上来说,已经不再仅仅是相互关联和链接的一组页面了。目前所谓的 AJAX 模型从根本上改变了用户对 Web 应用程序的体验,开发人员在设计和实现新的 Web 应用程序时,不得不考虑采用这一更新、功能更加丰富的模型。言归正传, AJAX 模型到底是怎么回事呢?

AJAX 这一相对来说比较新的术语,是“*Asynchronous JavaScript and XML*”的缩写。它是一个涵盖性术语,所描述的技术主要用于创建具有高度交互性和响应性的 Web 应用程序。什么是 AJAX 的精髓呢?早在 10 年之前,创建 Web 应用程序时不就要求具有“交互性”、“响应性”,并可以通过浏览器来部署吗?那么现在的 AJAX 在这些方面的创新是什么呢?

Internet 所取得的巨大成功使得人们对 Web 相关技术进步的渴望出乎意料地与日俱增。多年来,用户对功能更加强大的 Web 应用程序和服务的迫切要求,促使体系结构框架设计师和程序开发人员将越来越多的特色功能集成到服务器平台和客户端浏览器中。因此,传统的 Web 应用程序模式已不能满足功能的要求。在设计模式和编程模型上的彻底变化迫在眉睫。

在传统 Web 应用程序的开发方针和原则的基础上对平台进行改进和增强,已经不能应对最新技术需求对业界提出的挑战,这时需要真正的革命性变革才能解决问题。对下一代 Web 应用程序来说, AJAX 正是这种变革的化身,可能在未来 10 年之内它都会经久不衰。

从开发人员的角度看, AJAX 主要是指一组用于创建具有高度交互性 Web 应用程序的开发组件、工具和技术,可以提供更好的用户体验。按照 AJAX 规范, Web 应用程序与 Web 服务器交换的仅仅是数据而不是页面。从用户角度看,使用 AJAX 意味着向服务器发送和接收数据的速度更快,更主要的是大大减少了页面加载和刷新的负载。这样的结果是, Web 应用程序将趋向于传统的桌面 Microsoft Windows 应用程序,并具有后者的一些高级功能,如拖放、异步任务、快速的响应和非抖动的用户界面,以及其他众多减少用户麻烦的功能,为用户提供定时的进展反馈,发送大量的泥搭 *mashed-up* 译为混搭内容。(但是,这并不是说 AJAX Web 应用程序与桌面程序完全一样,它们只是支持一些更加桌面化的功能)。

在 AJAX 基本原理的启发下,已经出现了为数众多的组件和框架,它们针对特定的平台提供了一组功能,或许可以和现有的框架无缝地集成。Microsoft ASP.NET AJAX Extensions

为 ASP.NET20 平台提供了 AJAX 支持。在下一版本的 NET 框架中, ASP.NET AJAX Extensions 将会正式集成到 ASP.NET 和其他 Microsoft Web 平台和应用程序模型中。下一版本的 Microsoft Visual Studio(代码名称为“Orcas”)也将会为 AJAX 的特定功能提供动态设计的支持。

在本章中,我们将深入研究使 AJAX 大行其道的应用动机和驱动力,以及支持所有 AJAX 框架所需要满足的基本系统要求。

1.1 变革的过程

在大家不懈努力下,现在的 Web 正在经历一场划时代的变革,作为见证人和推动者,我们都愿意为这一有趣而独特的现象推波助澜。这不仅仅是听起来响亮的口号,Web 已经改变了人们对应用程序的概念。就在 8 年之前,那时大多数开发人员认为开发应用程序是件很严肃的事情,不能将其精简为一套脚本和标记代码的无序混合体。在上世纪 90 年代,开发应用程序的代价就是程序员辛苦劳动的血汗和无休止的调试周期。根据公众的、不太严肃的感觉,人们不会把荣誉和名利赠送给那些“真正”的写 Web 应用程序代码的程序员。



注意 虽然在 20 世纪 90 年代末,也设计和建造了许多 Web 站点,它们当中的一些在随后的几年中速度扩展,成为当今世界经济的中流砥柱,甚至已经改变了我们处理日常事务的方式。Google、Amazon、eBay 等等就是它们的代表。尽管如此,10 多年前创建这些应用程序的人们仍然是当时开发人员当中的先行者,或者只是些聪明的游戏业余爱好者而已。

从那以后,Web 的发展突飞猛进。虽然历经 10 年的 Web 发展在服务器端抽象应用技术上已经积累了丰硕的成果,但是最基本的基础结构(HTTP 协议和页面)一直没有发生变化。

最初的 Web 基础结构是非常简单、通用而有效的,因此,它是应用程序的 Web 模型快速成功的首要因素。下一代的 Web 应用程序模型仍将会基于 HTTP 协议和页面。但是,页面的内容和服务器端机制的功能会有所变化,将尽可能地像传统桌面 Windows 一样为用户提供丰富的使用体验为主要目的。



注意 正如在第 8 章“用 ASP.NET 创建 AJAX 应用程序实例”中讲述的细节, AJAX 应用程序虽然拥有众多优点但还是存在缺点。总的来说,选择 AJAX 应用程序而不是传统的 Web 应用程序是权衡利弊后的结果。AJAX 应用程序肯定会给用户提供连贯的反馈,绝不会因为等待远程操作而挂起用户界面。另一方面, AJAX 应用程序和桌面应用程序有所差别,它在图形处理、多媒体和硬件

控制方面的能力还比不上常规的(智能)客户端。最后, AJAX 应用程序是一种非常特殊的 Web 应用程序,可能还需要某种代码重构才能达到预期的性能和结果。

1.1.1 传统的 Web 应用程序

现在 Web 应用程序的工作原理是将用户填写的表单提交给 Web 服务器,然后再显示由 Web 服务器返回的标记(即 html)页面。浏览器和服务器之间的通信使用了传统的 HTTP 协议。众所周知,HTTP 协议是无状态的,这意味着每次请求之间是互不相关的,系统不会自动维护状态信息。(我们所熟知和使用的状态对象,ASP.NET 只不过是对服务器编程环境的一种抽象。)

在浏览器和 Web 服务器之间的通信都是通过“表单”完成的。从用户的角度看,变化是通过“页面”发生的。每个用户操作都会引发一个新的服务器请求,结果是下载并显示一个新的页面(或是当前页面经过修改的一个版本)。

让我们深入地研究一下这种模型,看看它有什么缺点,并搞明白需要一种新模型的原因。

1. 通过表单传送输入数据

在浏览器的地址栏中输入 URL 后,会显示用户请求的页面。最终的页面将是由 HTML 标记组成的,包含了一个或多个 HTML 表单。用户输入数据,再告诉浏览器将表单提交到一个用于处理操作的 URL。浏览器使用操作系统中提供的本地域名系统(DNS)解析器,把特定的 URL 转换为一个 IP 地址,并打开相应的套接字(socket)连接,这样,HTTP 数据包便会通过网络线路传输到指定的目的地。数据包中封装了表单和其所有字段的数据,Web 服务器会捕获请求,再将其转发到内部的相关模块进行处理。在处理完成后,会生成 HTTP 响应数据包,发给客户端浏览器的返回数据会封装在这些数据包中。

2. 通过页面得到输出

当生成一个对 aspx 资源的请求时,Web 服务器会将它传递到 ASP.NET 进行处理,之后再接收处理的 HTML 标记内容。生成的标记内容由传统的 HTML 页面标签组成,包括 <HTML>、<body> 和 <form> 等等。页面源代码将被嵌入在 HTTP 响应流中,并通过“多用途互联网邮件扩展”(Multipurpose Internet Mail Extensions, MIME)类型标记来告诉浏览器如何处理响应内容。浏览器会根据 MIME 类型来决定具体要做什么。

如果响应包含了一个 HTML 页面,浏览器会用一大段 HTML 标记内容来完全替换当前的页面。但当服务器正在处理请求时,原来的“旧”页面会被“冻结”,一直显示在客户端用户的