

/THEORY/IN/PRACTICE

O'REILLY®

Ajax Design Patterns

Ajax 设计模式

使用程序设计和可用性模式创建Web 2.0站点

Michael Mahemoff 著

杨仁和 译 ajaxcn 审校



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Ajax 设计模式

Ajax Design Patterns

[美] Michael Mahemoff 著

杨仁和

译

ajaxcn

审校

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是一本关于复杂 Ajax 应用的整体架构设计的图书。好的架构设计应该满足 KISS 的原则,容易理解和学习、开发效率高、容易维护和扩展。在各种必须要满足的约束条件之下,设计出一个最简化的架构是架构师不断追求的目标。幸运的是,经过国外高水平的 Web 开发专家的探索,这样的架构设计已经有了一套行之有效的模式。这些模式都是围绕着 REST 的思想来设计的。RIA+REST 将会成为继 Web MVC 之后,新一代 Web 应用的首选架构风格。

本书是国内出版的第一本深入探讨 REST 架构设计的书籍。书中的所有模式都有详细的讲解和丰富的代码,理论联系实际,确实是一本不可多得的好书。

0-596-10180-5 Ajax Design Patterns. Copyright © 2006 by O'Reilly Media, Inc. Simplified Chinese edition, jointly published by O'Reilly Media Inc. and Publishing House of Electronics Industry, 2006. Authorized translation of the English edition, 2006 O'Reilly Media Inc., the owner of all rights to publish and sell the same. All rights reserved including the rights of reproduction in whole or in part in any form.

本书中文简体版专有出版权由 O'Reilly Media, Inc. 授予电子工业出版社,未经许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字:01-2006-5603

图书在版编目(CIP)数据

Ajax 设计模式 / (美) 麦赫马夫 (Mahemoff, M.) 著; 杨仁和译. —北京: 电子工业出版社, 2007. 5

书名原文: Ajax Design Patterns

ISBN 978-7-121-03995-9

I. A... II. ①麦...②杨... 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 033293 号

责任编辑: 周 筠 晓 菲

项目编辑: 梁 晶

印 刷: 北京智力达印刷有限公司

装 订: 北京中新伟业印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 42 字数: 870 千字

印 次: 2007 年 5 月第 1 次印刷

定 价: 78.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。服务热线: (010) 88258888。

O'Reilly Media, Inc.介绍

为了满足读者对网络 and 软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权电子工业出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是在线出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》(被纽约公共图书馆评为20世纪最重要的50本书之一)到GNN(最早的Internet 门户和商业网站),再到 WebSite(第一个桌面 PC 的Web服务器软件), O'Reilly Media, Inc. 一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为O'Reilly Media, Inc. 紧密地与计算机业界联系着，所以O'Reilly Media, Inc. 知道市场上真正需要什么图书。

审校者序

传统的 DHTML 技术与 XMLHttpRequest、IFrame 等 Web Remoting 技术相结合之后，得到了服务器端强有力的支持，这使得 DHTML 技术如同插上了腾飞的翅膀，并孕育出了一种被称作 Ajax 的新技术。自从 Ajax 技术出现之后，对于其本质和如何以最有效的方式来使用它的研究一直在持续进行着，其目的是使 Web 应用获得在可用性、用户体验和开发效率等方面的最大利益。这一切的研究都是建立在现有实践的基础之上的。实践总是远远走在理论的前面，这是广大富有创造性、勇于开拓的 Web 开发者的功劳。互联网经济的复苏造就了 Web 2.0 的神话，Web 2.0 则促进了 Ajax 技术的发展。在 Web 2.0 网站中，对于 Web 应用的交互设计提出了各种苛刻的要求，这促使 Web 开发者不断去探索新的未知领域，探索基于 Web 标准的技术所能达到的极限。Web 开发中新的需求层出不穷，这为他们发挥自己的创意、实现自己的梦想创造了广阔的舞台。

今天，浏览器中的 JavaScript 通过 SVG、Canvas 等技术获得了全面的绘图能力，可以用来绘制图表和开发视频游戏。浏览器的最新发展还为 JavaScript 脚本提供了更大的本地存储空间，远远超出了 Cookie 的 4KB 数据量的限制。这非常有利于开发复杂的 Ajax 应用（例如 online 的字处理器），这些应用需要在客户端保存大量的状态。Ajax 应用可以得到 Flash 的协助，可以通过 Flash 的能力来播放多媒体内容。Ajax 应用还可以通过 HTTP 信息流（另外一个名称叫做 Comet）的方式与服务器通信，在获得了更好的实时性的同时，也消耗了更少的网络流量和服务器端资源。通过这种方式可以实现信息的实时推送，这种技术可以被应用在股票信息实时发布和即时通信等广泛的领域中。

今天的 Web 开发与 3 年前相比已经有了非常大的不同，新的事物仍在不断涌现。如果有一个见多识广、富有洞察力的人将这些技术系统地整理起来，对广大 Web 开发者开阔眼界，提高技能会有非常大的帮助。《Ajax 设计模式》这本书恰逢其时，刚好填补了这样一个巨大的空缺。在书中，作者 Michael Mahemoff 以最广阔的视野，最大努力，将当

今已经出现在 Web 上的各种基于 Ajax 的开发技术进行了一个全景式的展示，系统地将这些技术以模式的形式加以整理和编目，并辅以精心设计的代码。本书是对于 Ajax 技术成型近两年以来所取得成果的一个全面总结。所有的 Web 开发者都可以从本书中受益，学习书中所列出的众多的开发技术和开发套路，并尽快将这些优秀的技术应用在自己的项目中。

古龙的《多情剑客无情剑》（又名《小李飞刀》）是大家熟悉的武侠小说。其中有一个人叫做百晓生，这个人似乎无所不知无所不晓。他编写了一个兵器谱，对于江湖中最厉害的兵器进行了排序。小李飞刀排名第三，因此李寻欢又被称做小李探花。《Ajax 设计模式》的作者就是 Ajax 世界中的百晓生。他是著名的 Ajax 站点 AjaxPatterns 网站的站长，本书的内容是以在 AjaxPatterns 网站建立一年半以来所开展的讨论为基础的。

自从 GoF 的《设计模式》一书出版后，将各个技术领域中之行之有效的一些通用的解决方案总结为模式，成了一种对于该领域进行研究的最有效的方法。一个技术领域出现了大量的模式，说明对于这个领域的研究已经进入了高级阶段，这个领域的技术已经发展成熟。现在，对于 Ajax 领域的研究也进入了这样的一个高级阶段。不过 Ajax 技术距离完全成熟还很远，在近几年之内仍然会取得迅速的发展。Martin Fowler 在面向对象分析和设计（OOAD）领域做出了开创性的杰出贡献，Michael Mahemoff 对于 Ajax 领域也做出了类似的贡献。也许无法相提并论，但是他使得对于 Ajax 技术的研究纳入了理论化、系统化的快车道，非常有助于将来对于 Ajax 技术进行更加深入的研究。

基于 Web 标准的开发技术仍然会统治互联网，主要原因有两个：

1. 基于这些技术可以提供目前 Web 应用可获得的、最好的可用性。

国外对于 Web 可用性的研究已经有很多年，得到了很多行之有效的设计原则和最佳实践。W3C 对于提高 Web 应用的可用性也非常重视，在上世纪 90 年代就建立了 Web Accessibility Initiative (WAI, <http://www.w3.org/WAI/GL/>)。美国在 Web 应用的可用性方面还有相关的立法 (<http://www.section508.gov/>)。ISO 也正在制订 Web 可用性方面的标准 (<http://www.userfocus.co.uk/articles/ISO23973.html>)。

所以，Web 应用要达到很高的可用性，目前唯一的选择就是基于 Web 标准的技术。如果把这些研究成果和最佳实践统统扔掉，改用其他的 RIA 技术，都会影响 Web 应用的可用性。可用性与界面的华丽是两个不同的问题。

2. 这些技术对于搜索引擎是最友好的，便于进行搜索引擎优化。

假设我在网络上从事电子商务。如果采用 Web 标准之外的技术，很可能会严重影响我的网站对于搜索引擎的可视性 (visibility)，这会严重影响我的业务。因为目前绝大多数通用的搜索引擎都是围绕基于 Web 标准的技术而设计的，它们搜索 HTML 内容最为有效，如果是 Flash 内容或者其他内容，很有可能无法被搜索到。

所以我判断基于 Web 标准的技术仍然会统治互联网 (Internet)。而其他的 RIA 技术呢，

主要将占据企业内部应用 (Intranet) 的市场。

Ajax 技术将来的发展趋势我预测会有这些：

1. 各种 Ajax 组件库和框架逐渐成熟，提供了更多功能丰富的组件。使用这些组件做开发可以改善 Web 应用的可用性，并且可以大幅提高开发效率。
2. 各种 Ajax 组件库和框架会逐渐融合，并且改善它们之间的互操作性。
3. JavaScript 2.0 推出并且逐渐被广泛采用。其所提供的新功能对于基于 JavaScript 的大规模组件化开发非常有价值。
4. Ajax 开发工具逐渐成熟，出现功能全面的 IDE，支持 JavaScript 的代码补全、重构、调试、单元测试、部署等功能。
5. 浏览器本身也有可能实现一些将来被广泛应用的 Ajax 组件，例如 Tree、DataGrid 等。

作为《Ajax 设计模式》一书的审校者，我有幸先于国内的读者读到本书的内容。我的感觉就像是在森林深处发现了一个巨大的宝藏，阿里巴巴初次见到宝藏的那种感觉。我原先所理解的 Ajax 技术所能达到的范围一下子被拓宽了很多倍。书中大量有趣的技术，就像是一件件闪着璀璨亮光的珍宝，我都想尽快去尝试一番，感觉我似乎一下子变成了一个富翁。《Ajax 设计模式》一书是 Ajax 开发者的百宝箱，我们怀着虔诚的心情，将本书奉献给国内的 Web 开发者。希望读者在阅读本书的过程中能够获得与我们相同的体验，享受到巨大的快乐。Web 开发并不是一场苦功，而是一件很快乐的事情，让我们与国外的 Web 开发专家一道，尽力使它变得更加有趣。

本书由 ajaxcn.org 负责审校。李锐负责审校第 1~5、10、11 章和正式章节之外的所有文字，包一磊负责第 8~10 章，俞黎敏负责第 13~17 章，张祖良负责第 6、7、18、19 章。由于能力有限，难免会出现一些错误，敬请读者指正，我们一定会在后续的印刷中尽快改正。

李 锐

2007 年元宵节 于上海

审校者简介

李锐 (网名: dlee)

独立咨询顾问, Ajax 中国网站 (ajaxcn.org) 站长。热衷于 Web 表现层技术, 是国内 Ajax 技术和 Web 标准的热心推动者之一。gtalk: dlee.cn@gmail.com

包一磊

时任上海锐道 (BSTEK) 公司的 CTO。具有多年 Web 表现层技术的研究和开发经验, 领导开发了上海锐道的核心产品——dorado 展现中间件。E-mail: benny.bao@bstek.com

俞黎敏 (ID:YuLimin 网名: 阿敏总司令)

现任萨蒂扬软件技术 (上海) 有限公司项目经理, 主要从事电力、电信行业的系统开发与管理。积极参与开源社区工作, 担任 Dev2Dev、CSDN、CJSDN、Matrix、JavaWorld TW 等 Java 论坛版主。

张祖良

长期专注于 Web 领域技术, 曾担任系统工程师, 项目经理等角色, 有 php 和 .Net 开发经历, 现在专注于 J2EE 领域企业级应用开发。

设计模式索引

Ajax 应用 页码 63

一种能运行在任何现代 Web 浏览器内的表现能力丰富的应用。

Ajax 存根 页码 196

使用能让浏览器脚本直接调用服务器端操作的 Ajax 存根框架，而不需要担心 XMLHttpRequest 和 HTTP 传输的细节。

浏览器端缓存 页码 314

在浏览器端的缓存里保留服务器响应的结果。

浏览器端模板 页码 287

使用内嵌 JavaScript 生成模板，并要求浏览器端框架将它们展示成 HTML。

浏览器端测试 页码 608

建立浏览器端 JavaScript 组件的自动化测试。

浏览器端 XSLT 页码 280

应用 XSLT 将 XML 消息转换成 XHTML。

调用跟踪 页码 226

追踪 XMLHttpRequest 调用，从浏览器到服务器，再从服务器返回浏览器。

跨浏览器组件 页码 306

建立跨浏览器兼容的组件，让程序员重用它们，不必考虑个别浏览器的特质。

跨域代理 页码 262

建立负责代理与中介的 Web 服务，促成浏览器与外部域之间的沟通。

DOM 查看 页码 592

使用 DOM 查看工具，探索动态的 DOM 状态。

数据表格 页码 380

表现能力丰富的表格，并支持常见的查询函式。

调试 页码 589

使用 JavaScript 调试器诊断问题。

直接登录 页码 533

代替基于表单的提交，以 XMLHttpRequest 调用为用户做认证，并且对浏览器里的数据进行哈希编码 (hashing)，以提升安全防护。

显示变形 页码 73

通过改变 DOM 里的样式与值 (例如 text 和 color 属性)，改变页面元素。

分布式事件 页码 251

使用事件机制让对象保持同步。

拖曳与放置 页码 426

提供拖曳与放置机制，让用户直接重新安排页面上的元素。

分级下拉菜单 页码 372

为了让用户在层次结构里找到条目，提供动态的分级下拉菜单。

明确提交 页码 245

代替随着每个浏览器事件自动提交，要求用户明确请求它；例如，依按钮点击提交。

胖客户端 页码 343

遵循只有在浏览器没有其它方法达到相同效果才执行远程调用的原则，建立丰富的、以浏览器为基础的客户端。

推测估计 页码 329

代替从服务器请求信息，改为进行合理的推测。

HTML 消息 页码 201

让服务器生成将要被显示在浏览器里的 HTML 片段。

HTTP 信息流 页码 122

为响应长时间存活的 HTTP 连接，将服务器数据以流的形式传送。

心跳检测 页码 557

让浏览器定期上传心跳检测消息，表明应用还在浏览器内，用户仍然活跃。

突出显示 页码 512

通过以一致的、能吸引注意力的格式展示元素，强调它们。

防主机端保存 页码 540

以加密形式保存敏感数据，客户端只提供绝不会被提交给服务器的密码短语，存取及操作它。

IFrame 调用 页码 116

使用 IFrame 进行浏览器-服务器通信。

JSON 消息 页码 218

以 JSON（JavaScript 对象表示法）格式，在服务器与浏览器之间传送消息。

延迟注册 页码 518

当正式注册被延迟时，随着与互动的进行，逐步积累用户信息。

动态命令行 页码 409

在命令列接口里，监控编写中的命令，并动态修改接口，支持互动。

动态表单 页码 418

通过完整的互动，验证及修改表单，而不是等待明确的提交。

动态搜索 页码 401

随着用户构建并且使查询条件更加精确，持续显示全部的有效结果。

日志 页码 586

为你的 JavaScript 提供消息记录。

可改变的内容 页码 444

建立由可改变的内容区块所构成的页面——一小块、一小块可以在页面里编辑的内容区块。

微链接 页码 454

提供在现有页面上打开新内容的微链接，而不是加载新的页面。

多阶段下载 页码 336

将内容下载分解成多个阶段，让较快速和重要的内容先到达。

按需下载的 JavaScript 页码 133

随需要下载并执行 JavaScript 片段。

一秒钟移动 页码 505

将元素从一个位置逐步移到另一个位置，或者暂时移动它，以传达事件已经发生的事实。

一秒钟变形 页码 498

当页面元素经历值的改变或其它重要事件时，动态变化它的形状约一秒钟。

一秒钟聚光灯 页码 488

当显示元素经历值的改变时——或任何其它显著的改变——动态增加它的亮度约一秒钟。

页面重新布置 页码 83

通过 DOM 操作，增加、移除、移动、及覆盖元素。

定期刷新 页码 231

浏览器定期发出 XMLHttpRequest 调用，获取新信息；例如，每隔 5 秒调用一次。

纯文本消息 页码 207

以纯文本格式在服务器和浏览器之间传送简单消息。

弹出框 页码 438

使用暂时性的弹出框（出现在标准内容之“前”的内容区块），支持快速任务与查询。

Portlet 页码 460

引进 Portlet——具有独立对话状态的独立内容区块。

预先获取 页码 322

预测可能的用户动作，预先获取内容。

进度指示器 页码 365

指出服务器调用的进度。

基于 REST 的服务 页码 174

根据 REST 原则暴露 Web 服务。

RPC 调用 页码 191

以 RPC（远程过程调用）暴露 Web 服务。

富文本编辑器 页码 386

实现具有易于使用的格式化功能和所见即所得的显示的富文本编辑器小组件。

富媒体插件 页码 161

使用富媒体插件，让你的应用“比 Ajax 更 Ajax”。

计划 页码 154

使用 JavaScript 定时器为动作安排执行的计划。

服务器端代码生成 页码 297

从服务器端程序代码自动生成 HTML 和 JavaScript。

服务测试 页码 612

建立自动化的 Web 服务测试。使用 HTTP 客户端与服务器互动，就像是浏览器平常做的那样。

模拟服务 页码 604

根据模拟真正服务的“伪造”Web 服务，开发浏览器应用。

滑动条 页码 358

提供滑动条，让用户在某个范围内选取值。

小精灵 页码 430

以小精灵加强显示——也就是，小的、灵活的、像图标一样的内容区块。

状态区域 页码 467

以只读的状态区域，报告当前和过去活动的状况。

提交数量控制 页码 238

代替针对每个 JavaScript 事件进行提交，在基于浏览器的缓冲区内保存数据，并以固定间隔自动上传它们。

输入提示 页码 393

根据用户当前的输入，建议他们可能完成的单字或习惯用语。

系统测试 页码 615

建立自动化测试，模拟用户行为并验证结果。

超时机制 页码 546

在一段闲置时间之后，让应用产生超时，并选择性地通知服务器。

流量嗅探 页码 598

通过嗅探 HTTP 流量，诊断问题。

唯一 URLs 页码 564

为重要的应用状态提供唯一的 URL。

更新控制 页码 472

让用户控制更新的速度与标准。

用户动作 页码 143

使用事件处理器，在 JavaScript 里处理大多数的用户动作。

虚拟工作空间 页码 476

提供参考到服务器端工作空间（workspace）的浏览器端视图（view），让用户浏览整个工作空间，就好像它被保存在本地。

Web 服务 页码 92

以细粒度的、易用的 Web 服务，暴露服务器端的功能。

XML 数据岛 页码 274

将 XML 响应保留为 XML 数据岛——在 HTML DOM 里的节点。

XML 消息 页码 211

以可扩展标记语言（XML）的格式，在服务器和浏览器之间传送消息。

XMLHttpRequest 调用 页码 98

使用 XMLHttpRequest 对象，进行浏览器-服务器通信。

序

对于想要知道 Ajax 正被如何应用到现实世界的开发者、设计者和经理人,《Ajax 设计模式》无疑是为他们量身打造的一份参考资料。Ajax 是一个新名词,描述行为丰富、类似桌面应用的 Web 应用,这些应用执行在标准的 Web 浏览器上,而不需要安装任何特别的插件。它的风行与普及,归功于一些众所瞩目的 Ajax 应用,例如 Gmail、Google Maps 和 37signals 的 Basecamp。在企业应用中, Ajax 也越来越受欢迎。

本书的模式分成 4 类。基础技术 (Foundational Technology) 模式概述了 Ajax 开发所需要的原始技术,例如 XMLHttpRequest 对象。编程 (Programming) 模式探索开发者所发现的一些技术,这些技术可以确保他们的 Ajax 应用易于维护且性能良好。功能性和可用性 (Functionality and Usability) 模式讨论将在 Ajax 应用中遇到的,以及 Ajax 使其成为可能的用户界面。最后,开发 (Development) 模式阐述了用来监控、调试,以及测试 Ajax 应用的过程。

谁该阅读这本书

谁该阅读这本书

如果想要达到以下目的，你应该阅读这本书：

- 了解 Ajax 能做什么，不能做什么，以及正如何被使用在现实世界里。
- 加速理解 Ajax 的核心技术，例如 XMLHttpRequest、DOM，以及 JSON 等。
- 发现开发者用来产生高质量的 Ajax 架构、改善性能，以及改善可用性的模式。

因为模式本身就是用来参考的，各种背景的读者都可以理解。想在 Ajax 领域快速提升的开发者，可以从教程的章节（第 2 章）及基础技术模式开始。已经有些 Ajax 开发经验的人，可以从编程模式和开发模式中获益良多。非技术领域的人，可以以高层的观点，特别是考察功能性和可用性模式以及它们在真实世界中的例子，来看看 Ajax 能做什么。

为了领会书中的技术性讨论与程序代码范例，程序员应该对网页开发有一定的经验——HTML、表单提交、服务器端脚本，等等。读者最好也了解一些 JavaScript，因为这本书不是来教你这门编程语言的。但是本书在写作时，作者确实考虑到了很多读者对 JavaScript 只是基本的了解。

Ajax 主要是关于发生在浏览器里的事，因此本书假设你不知道任何特定的服务器端环境。在那些涉及服务器端代码的地方，范例基本上都是以 PHP 作基础，并且用语言中立（language-neutral）的术语来作解释。

谁不该阅读这本书

谁不该阅读这本书

如果你没有从事过任何网页开发的工作，在一头栽进 Ajax 和这些模式之前，或许应该找些基础概念的介绍来看看会更好。

Ajax 开发涉及大范围的技术领域，包括 HTML、XML、CSS、JavaScript 和服务器端开发。这本书将帮助你理解 Ajax 与这些技术的每一项是如何关联的，通常是如何相结合的。但是，如果你指望彻底学习这些技术中的任何一项，或者你正在寻求有关浏览器细节的参考，建议你查阅更加专业的参考资料。

关于范例

关于范例

本书的所有范例（教程代码与 Ajax 模式的代码）都可以从 <http://ajaxify.com/run> 上下载。它们已经被开发成与新近版本的 Firefox 和 IE 兼容的代码，并且在 Firefox

1.5 和 IE 6.0^{注 1}上通过了测试。多数服务器端代码需要 PHP 5 或更新的版本,并在 PHP 5.0.4 上也通过了测试。PHP 程序代码应该运行在任何能执行 PHP 脚本的标准 Web 服务器上;Apache 1.3.33 被用于本书范例的测试。Wiki 的演示需要 MySQL,并且在 4.1.14-max 的版本上能通过测试。关于如何安装演示的信息,请参阅附录 B。

浏览器细节

实际上对 Ajax 的目的来说,所有的 Ajax 模式都能在任何现代的标准浏览器上实现,讨论的焦点应侧重于一般性架构与可用性的问题,而不是浏览器的细节。但是,可移植性(portability)的问题确实出现在了某些例子中,并且在对实现模式具有关键性影响的地方也作了说明(如同本书第二部分,基础技术模式的某些主题中所做的)。为了不模糊焦点,多数讨论仍局限于 IE 和 Firefox;描绘每个浏览器和版本的风格与手法,超出了模式所要讨论的范围,对此,建议你参考更加专业的资料。

本书的组织

第一部分,序曲

最初几章是所有模式的序曲。Ajax 简介(第 1 章)概述了 Ajax 现象并讨论当前的趋势。由模式引导的教程(第 2 章)是 Ajax 与 Ajax 模式的教学课程。Ajax 模式背后的设计理念则在 Ajax 设计:原则与模式(第 3 章)中讨论,并伴随了模式本身的介绍。

第二部分,基础技术模式

基础技术模式描绘了任何 Ajax 应用的核心“建造模块”(building blocks)。Ajax 本身是一种模式,因此作为第一个模式,Ajax 应用(第 4 章)扮演了所有 Ajax 模式的基础。接下来几章探讨创建 Ajax 应用所涉及的 3 项核心技术。显示操作(第 5 章)讨论用户界面的重绘,Web Remoting(第 6 章)整理了几种与服务器通信的做法。动态行为(第 7 章)讨论事件——所有衍生自用户或定时机制的事件。最后,扩展技术(第 8 章)解释如何以非标准的技术来扩展 Ajax 应用。

第三部分,编程模式

编程模式聚焦于软件的技术性质量,特别是可维护性、稳定性,以及效能。这部分的第 1 个章节,即第 9 章,提供了几种设计 Web Service 的策略。此外,与 Web Remoting 相关,浏览器-服务器对话(第 10 章)探讨浏览器与服务器间的信息流,例如,同步技术。DOM 组装(第 11 章)涵盖几项策略,处理继服务器响应之后的

注 1: 确保对所有主要的浏览器(例如 Safari, Opera)完全兼容会很好,特别是对于产品级别的系统而言。但是在这里,个别的演示并未特别考虑可移植性,每个演示主要是针对特定的 Ajax 概念做介绍。

DOM 组装 (DOM population)。代码生成与重用 (第 12 章) 包含了几个实现可维护性与可移植性 (portability) 的通用编程模式。最后, 性能优化 (第 13 章) 不仅讨论了提高更新与通信的速度, 也讨论了为无法避免的延迟优化用户的体验。

第四部分, 功能性与可用性模式

功能性和可用性模式着重于 Ajax 应用的可用性。小组件 (第 14 章) 介绍一些被整合进很多 Ajax 界面的小组件。页面架构 (第 15 章) 讨论了较高层次的观点, 其中的焦点是页面布局、内容分解, 以及暴露服务器端的内容。随着 Ajax 的普及, 因特网正经历一场视觉效果的提升, 这对传统的桌面应用系统是前所未有的; 最常见的视觉效果将在“视觉效果” (第 16 章) 中描述。这个部分的最后一章, 功能性 (第 17 章) 讨论了 Ajax 使之成为可能的几种新功能。

第五部分, 开发模式

开发模式不是你在 Ajax 应用中看到的“具体事物”——不像前面几个部分中的模式——而是能够用来辅助你开发的过程。诊断分析 (第 18 章) 协助进行疑难排查 (troubleshooting), 以及监测 Ajax 应用程序的运行状态。测试 (第 19 章), 在各个不同的层次上协助进行测试工作, 而且是建立在随应用的演进而逐步测试的敏捷实践 (agile practice) 的基础之上的。

第六部分, 附录

共有 4 个附录。附录 A, 最大的附录, 是一个 Ajax 相关框架和库的清单, 其中很多对实现本书中的模式大有帮助。附录 B 是一组安装代码范例的说明。附录 C 介绍模式的一般性理论。最后, 附录 D 列出了本书所参考的相关资料。

本书的格式约定

斜体

用来表示模式名称。

等宽字体

用来表示程序范例或片段。

等宽粗体

用来表示重要的代码。

“浏览器”这个词, 是浏览器端整个系统的简化表示——不只是浏览器应用 (例如 Firefox), 还包括运行在上面的 Web 应用 (例如 Google Maps)。当特指浏览器应用时, 通常会把它称为“Web 浏览器”——无论如何, 从上下文来看, 这是相当明显的。同样地, “服务器”或“服务器端”指的是服务器上的所有东西——操作系统 (例如 Linux), Web 服务器 (例如 Apache), 以及运行在上面的特定于应用的脚本。

“ECMAScript”和“JScript”这两个词并未出现在本书中，“JavaScript”被用来涵盖这些名称。

“类”这个词，被用来描述以类似面向对象系统中类的方式使用的 JavaScript 函数。

出现在每个模式里的第一个图形，是该模式的图解。

本书所使用的公共首字母缩略词如下所示：

- CSS：层叠式样式表（Cascading Style Sheets）。
- DHTML：动态 HTML（Dynamic HTML）。
- DOM：文档对象模型（Document Object Model）。
- HTML：超文本标记语言（Hypertext Markup Language）。
- IE：微软 Internet Explorer。
- XML：可扩展标记语言（eXtensible Markup Language）。

代码范例的格式约定

设计图解基于统一建模语言（UML）表示法，但在多数情况下，这并不是很正式。

在代码范例中，.phtml 后缀用于输出 HTML 的 PHP 文件；所有其他的 PHP 文件（例如商业逻辑模块）则以 .php 结尾。

如第 2 章所述，`$()` 是 `document.getElementById()` 的别名，用来减少代码的凌乱程度（受 prototype 库启发——见 <http://prototype.conio.net/>）。

为清晰起见，某些代码范例重新进行了格式化。