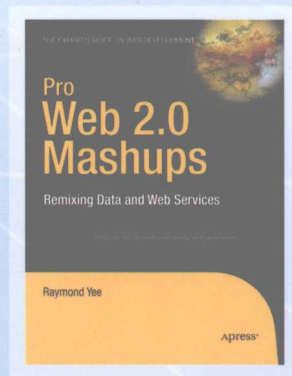


Pro Web 2.0 Mashups Remixing Data and Web Services

Web 2.0 Mashup 开发实战

[加] Raymond Yee 著
唐扬斌 等译

- Amazon全五星盛誉著作
- 深入揭示Web 2.0核心技术——Mashup（混搭）
- 兼顾PHP、ASP.NET等各种服务器的开发技术



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书 Web开发系列

Pro Web 2.0 Mashups Remixing Data and Web Services

Web 2.0 Mashup 开发实战

[加] Raymond Yee 著
唐扬斌 等译

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

Web 2.0 Mashup 开发实战 / (加) 伊 (Yee, R.) 著;
唐扬斌等译. —北京: 人民邮电出版社, 2009.4
(图灵程序设计丛书)
书名原文: Pro Web 2.0 Mashups: Remixing Data and
Web Services
ISBN 978-7-115-19697-2

I. W… II. ①伊…②唐… III. 主页制作—软件工具
IV. TP393.092

中国版本图书馆CIP数据核字 (2009) 第013522号

内 容 提 要

Mashup (混搭) 是 Web 2.0 核心技术之一, 本书讲解了使用 API、Web 服务、Ajax、Web 标准以及多种服务器端语言来创建混搭所需要的全部知识, 同时探讨了混搭的技术走向和发展趋势。书中还提供了丰富的混搭应用示例。

本书面向具有一定 HTML、CSS、JavaScript 和某种服务器端语言基础的 Web 开发人员。

图灵程序设计丛书

Web 2.0 Mashup开发实战

- ◆ 著 [加] Raymond Yee
译 唐扬斌 等
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>
北京铭成印刷有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 31.5
字数: 843千字 2009年4月第1版
印数: 1—3 000册 2009年4月北京第1次印刷

著作权合同登记号 图字: 01-2008-5239号

ISBN 978-7-115-19697-2 / TP

定价: 69.00元

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

反盗版热线: (010)67171154

译者序

任何一本书想要做到面面俱到都不易。但拿到本书时，我却觉得这确实是一本内容相当全面的教材。翻开目录，XML、JavaScript、SOAP、Ajax……各种与Web应用和开发相关的词都跃入眼帘。是的，从混搭的理念出发，我们就是要把所有相关的东西整合在一起，给人一种耳目一新的感觉。

从Web 2.0的大行其道，到人们开始对因特网与Web能力进行的反思，各种思想的汇聚（这是混搭的核心）一直都是技术探索和创新的动力。也许有一天，正如作者所言，我们所生活的世界里“看不到了”混搭，这并不是因为混搭已不存在，而是因为它无处不在，我们习以为常了。

无论是科技的发展、文化的进步还是思想的创新，混搭、混编或集成都属于人类活动的核心内容。

本书不是对所有Web应用技术的简单介绍，而是重点剖析了一些与混搭的创建和创新紧密相关的技术。作为一个非专业的技术人员，你也许会担心自己无力涉足混搭领域，只能享受他人的成果。本书所要传达的第一个重要信息就是，混搭并非高不可攀。

本书第一部分对于任何一个经常浏览网页的人而言，都是一份切实可行的混搭指南。而作为专业技术人员，特别是关心Web应用开发的程序设计人员，本书的核心——Web API编程——将为你展示可使用混搭的各种情况。虽然不能全部详细阐述，但它所解决的每一个问题和展示的每一种思想都会为深入研究和学习带来启发，成为你进一步了解和应用混搭的基础。对混搭的技术走向和发展趋势的把握也是本书的一个关注点。作者作为从事软件集成多年的专业人士，不仅从内部对混搭进行了详细的剖析，而且也描述了其外部情况（可用的资源、工具及其发展方向）。相信任何阅读本书的人都会从中受益，并由此在Web应用的技术和发展道路上跨上新的台阶。

“集成创造价值”——正如作者所言，从事混搭的人能够从混搭中获得乐趣，同时也应富有冒险精神。我们对Web世界的探索和发现从来都不曾停止，希望你可以通过本书发现一个新的更美好的Web世界。

本书主要由唐扬斌翻译。参与本书翻译的还有张鼎、叶俊、颜炯、富弘毅、何小威、奚丹、陈钢、王沛、陈小文、闫志强、薄建禄、林龙信、邓彬、胡季红、邝祝芳、汤亚锋、焦贤龙等。肖国尊负责本书译者确定、翻译质量和进度的控制，以及翻译思想的指导，在此表示衷心感谢。

鉴于译者水平有限，错误在所难免，还望读者谅解并不吝指正。

译者

2008年年底于湖南长沙

前言

有多少次，当你看到一个Web站点时会禁不住说：“这正是我想要的，要是……”要是能将其中的统计数据和公司赢利预测结合起来；要是能抄下那些餐厅地址，然后标注在地图上……有多少次你可以只点一下鼠标，而无需敲键盘，就可以将音乐会的日期添加到日历中？有多少次你曾经期望将个人数字世界中的组成部分（电子邮件、字处理器文档、照片、搜索结果、地图、演示文稿）统统无缝地组合在一起？毕竟，它们都是可重塑的数字信息，难道它们不应该结合在一起吗？

事实上，我们使用的所有这类数据、Web站点和应用程序都可以在系统中结合起来。本书的目的就是告诉你如何通过信息的组合来创建自己的混搭，从而发掘出众多潜在的联系；使整个Web为你所用。按维基百科的解释，混搭（mashup）是指“将多个来源的内容无缝地组合成一个集成体验”^①的Web站点或Web应用。学习如何获取来自不同Web站点的内容，并将其一同置于新的集成界面或程序中加以应用（无论是为你自己还是为别人），正是本书的中心内容。

不妨来看几个例子，从中了解一下目前人们是如何通过混编数据和服务来创建新的有价值的应用的。

- ❑ Housingmaps.com把Google Maps和来自Craigslist.com的房屋租售列表组合在了一起。值得一提的是，发明它的是独立程序员Paul Radamacher，而不是Google或Craigslist。Housingmaps.com为Craigslist增添了新的功能，能够在地图上显示出具体的房屋所在的位置，而不是某个地区房屋租售的所有信息^②；
- ❑ GMiF (Google Maps in Flickr) 利用Greasemonkey脚本将Flickr图片、Google Maps、Google Earth和Firefox浏览器联系在了一起^③；
- ❑ Library Lookup书签是一个基于JavaScript的书签小工具（bookmarklet），能够将Amazon.com与你本地的图书馆目录连接起来^④。

为了创建属于自己的混搭和对个人Web站点进行定制，在本书我们将细致地研究以上这些例子。当然，本书还会提供更多大大小小的示例。我们可以通过信息混编来解决无数的具体问题。以下是你将通过本书了解的一些内容。

- ❑ 在你的本地图书馆中迅速找到Amazon.com上的某种图书；
- ❑ 利用Yahoo! Pipes从多个新闻来源综合出一个新闻提要（feed）；

① [http://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)) 或 http://en.wikipedia.org/w/index.php?title=Mashup_%28web_application_hybrid%29&oldid=98002063。

② <http://housingmaps.com/>。

③ <http://webdev.yuan.cc/gmif/>。

④ <http://weblog.infoworld.com/udell/stories/2002/12/11/librarylookup.html>。

- ❑ 通过单击某个按钮将Flickr照片发布到博客;
- ❑ 在Google Maps和Google Earth中显示照片;
- ❑ 利用Firefox的特殊扩展来学习如何对Google Maps编程;
- ❑ 使用Greasemonkey在网页中插入额外信息;
- ❑ 在地图上标出来自自己感兴趣的新闻源（比如《纽约时报》）的报道;
- ❑ 使自己的Web站点变得可混编，从而使得他人能够利用你提供的内容创建混搭;
- ❑ 由你个人来自Web的事件日历创建一个Google日历;
- ❑ 利用在线存储（S3）存取文件;
- ❑ 利用Amazon.com的期望表（wishlist）创建一个在线电子表格;
- ❑ 识别和处理网页内嵌的数据;
- ❑ 将Web上列出的某个事件加入自己的日历以及通过单击发邮件将其传给他人;
- ❑ 在Web应用中建立搜索功能;
- ❑ 根据Web站点的格式要求重新发布Word文档。

混搭是当今的热点主题之一，它非常吸引人，因为它能使你成为一项共同事业或活动中的一份子。混搭不仅有趣，而且常常具有教育意义。看到那些熟悉的事物被放在一起，创造出了新的、超出简单相加的结果（对有些混搭而言，也不一定要严格做到这样），你的心情一定是相当愉快的。此外，混搭是非常强大的，它可以令你事半功倍。它们也许并不会永远都用到，但我们总可以从中获得自己需要的东西，而不需要投入过多精力。

Web 2.0 运动

Web 2.0的大行其道是导致混搭如此流行的一个重要原因。在Tim O'Reilly的“[What is Web 2.0?](http://www.oreil.lynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html)”（什么是Web 2.0?）^①一文中，明确地提出了混搭一词（他文中的说法是“可混编数据源”和“混编的权力”）。在此之后，我们已经通过混编/重用数据、Web服务和微应用来创建复合应用，将“Web 2.0 技术/思维模式”推进了一大步。最近的进展为用户重组数字内容和服务提供了更好的支持，相关的进展包括：

- ❑ 提高XML数据源和数据格式在商业、个人以及一般客户端应用（包括办公套件）中的可用性;
- ❑ XML Web服务的广泛部署;
- ❑ 对数据混编/混搭的广泛关注;
- ❑ Ajax以及基于JavaScript的微件和微应用的可用性;
- ❑ Web浏览器演化所带来的更好的可扩展性（例如，Firefox扩展和Greasemonkey脚本）;
- ❑ “用户创作内容”或“领先用户创新”模式的爆炸性增长;
- ❑ 因特网作为平台（“Web 2.0”）的概念深入人心;
- ❑ 宽带接入的增长。

这些进展已经使创建混搭由一项挑战变成了近乎主流的技术。开始进行混搭并不难，但我们还是需要了解很多东西，此外，从事混搭的人应该有学习兴趣，并有点冒险精神。

混搭技术会一直处于技术前沿吗？显然不会。这并不是因为这种技术本身将变得过时，而是因为

^① <http://www.oreil.lynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>。

它终将成为电子化社会里随处可见的日常技术。

此外,混搭反映了人类愿望中深层甚至是最深层的趋势。随着信息数量的增加、信息质量的提高,以及信息多样性的增长,用户期望获得一些新的工具,帮助他们有效访问和管理错综复杂的信息流。对很多用户而言,最终能令他们满意的信息环境,不仅应该可以支持无缝访问所有数字内容资源、处理各种内容类型,还应该提供针对这些内容的各类软件服务。例如,一群博主^①这样描述他们理想中的下一代博客工具^②:

博主们需要的是一种能够将他们所想到的任何东西,以任意格式、通过任意工具、在任意地点进行博客发布的真正简单的工具。文本只是开始,博主们要求将数据的类型延伸到更多不同的媒体,包括音频、图片和视频这些特征丰富且智能化的应用。关于内容的输入,拥有一个对话框已经被很多博主视为最初级的阶段,从利用可视化工具轻松抓取博主所看到的、听到的或正在阅读的内容,到使用未来期望出现的新型日志工具的用户界面,都应该成为可能的途径。

混搭所要打造的正是这样一种备受瞩目的针对各种数据和工具的访问途径和集成方式,它所针对的数据和工具显然不仅仅局限于博客应用,而是包含了用户和内容交互的方方面面。

全书概览

本书的中心问题是,如何帮助开发者和非技术出身的最终用户实现数据和因特网服务的重组,以创建对其个人或他人有用的新工具。尽管本书主要关注XML、Web服务和各类Web应用,但也会涉及桌面应用程序和操作系统等相关的内容。

本书结构

以下是本书各部分及各章节的内容简介。

□ 第一部分介绍不需要编程实现的混搭,同时介绍对应用进行分解的技巧,为混编做好准备。

- 第1章详细分析一组混搭/混编案例(例如,Housingmaps.com、Flickr中的Google Maps以及LibraryLookup书签小工具),带你认识一些本书后续将会不断回顾的内容,包括一般的混搭及其相关模式。
- 第2章分析Flickr(以之作为主要的扩展示例)何以成为一个卓越的混编平台。在此基础上,介绍如何混编某个具体应用,以及有哪些影响可混编的特性。我们将对比Flickr与其他一些混编平台(例如del.icio.us、Google Maps和Amazon.com)。
- 第3章介绍标签机制。标签机制允许用户为图片和Web站点(几乎可以是Web上的一切)附加文字。它就像胶水,将Web站点的内外粘合起来。我们将学习如何在Flickr、del.icio.us和Technorati中使用标签,研究如何创建以标签为中心的有趣混搭,观察人们如何“切入”(hack)标签系统以创建各式各样的数据库,并讨论标签与其他分类系统的关系。
- 第4章介绍RSS和Atom两种也许是应用最广泛的XML语言。一方面把它们作为一直实现混编

① 拥有博客的人。——译者注

② http://www.cadence90.com/blogs/2004_03_01_nixon_archives.html#I07902918872392913。

的行之有效的技术，另一方面也将其作为从更一般的角度学习XML语言的具体途径。其中不能错过的，是关于各种RSS/Atom相关格式及其在信息混编中重要地位的介绍。本章还介绍了如何应用Yahoo! Pipes实现提要的过滤和综合。

- 第5章从Flickr与博客的集成应用出发，介绍博客、wiki及其可编程性。相对于前述通过标签或直接基于RSS的面向数据的混编，博客所提供的是一类叙述式的混编，因此对博客进行集成是一个重要的主题。本章最后对wiki集成进行了简单介绍。
- 第二部分通过不同类型的范例，重点介绍如何使用类型丰富的基于Web的API。
 - 第6章详细研究Flickr。除了作为本书第一部分中不需要编程的混编技术的示例，Flickr同时也为学习XML Web服务提供了良好的背景。本章将介绍如何使用Flickr的API。首先介绍如何实现简单的Flickr API调用，然后了解可用的所有类型的调用，最后讨论其中的认证机制。
 - 第7章介绍不同API提供程序的异同，特别是对Flickr与其他系统进行了比较；综述目前可用的服务类型；讨论如何理解API的多样性。具体介绍如何调用REST、XML-RPC以及基于SOAP的服务。本章将会提到一些站点，例如Programmableweb.com，它们为不同API及其面临的应用挑战提供了相关的文档。
 - 第8章介绍另一大类可混编的Web应用，即基于JavaScript的微件（其中许多是Ajax应用）。本章通过Flickr和其他应用的示例，对比旧风格的Web应用和Ajax方式。介绍Yahoo! UI Library，一个JavaScript微件库，演示如何编写微件。此外，还将介绍如何使用Firebug Firefox扩展，以及通过JavaScript Shell了解JavaScript。最后，介绍Greasemonkey的使用。
- 第三部分，也即本书的核心，讨论如何利用前两部分介绍的技术来创建真正的混搭。
 - 第9章通过研究一系列具体的应用问题，分析混搭与相关API之间的关系，这些应用问题都是混搭可以发挥作用的典型场景。介绍如何通过各种API的组合跟踪图书、房产、机票价格乃至正在发生的时事信息。介绍如何使用Programmableweb.com分析这些问题。
 - 第10章通过一个具体示例，从零开始一步步地介绍如何编写混搭。该例为带有地理标签的Flickr照片与Google Maps的混搭，其中将先后用到Google Maps和Google Mapplets的API。
 - 第11章讨论一些已知的可用于创建混搭的工具，利用它们来创建混搭比使用传统的Web编程技术更加方便。详细介绍了这些工具的代表之一——Google Mashup Editor，然后简要综述其他工具。
 - 第12章将如何使用数据和API的讨论转向如何创造它们。这一章为那些希望使自己站点可被混搭的内容生产商提供了指南。换句话说，它回答了这样一个问题：作为内容生产商，你该如何使自己提供的数字内容能更加有效地被其他用户和开发者们混编和混搭？
- 第四部分讨论如何利用前三部分提供的核心概念框架，对特定类型的应用程序进行混编和集成。
 - 第13章通过基于地图的混搭示例，介绍流行的在线地图和虚拟地球应用。首先介绍如何在不需要编程和使用相关数据交换格式（GeoRSS和KML）的前提下创建地图。然后介绍一些各不相同的API，包括Google Maps、Yahoo! Maps和Microsoft Maps。在此过程中，介绍美国和美国之外地址的地理编码。最后讨论Google Earth，它和KML的关系，以及如何使用KML显示Flickr照片。
 - 第14章介绍社会化书签机制如何应对一个基本的挑战，即如何保存从Web上找到的内容？这

些内容一般情况下是一些URL，但在此我们还要面对一些其他类型的资源，例如图像和数据集。社会化书签的有趣之处不仅仅在于构建在这些系统内的可扩展性/可混编性，也在于它所提供的对其他系统的深入分析。本章将介绍一组社会化书签系统及其API，同时对这些系统间的互操作进行了讨论。最后，介绍如何创建一个Flickr和del.icio.us的混搭。

- 第15章介绍如何不通过编程（使用iCalendar和XML提要）读写在线日历中的数据，如何编写个人日历（使用Google Calendar和30boxes），以及如何编写个人事件聚合器的API（使用Upcoming.yahoo.com和Eventful.com）。最后介绍一个公共事件日历与Google Calendar的混搭。
- 第16章综述正在成长、具有巨大潜力的在线存储解决方案，同时介绍Amazon S3的基本用法。
- 第17章介绍如何对ODF和OpenXML进行解析，演示如何创建ODF和OpenXML文档，介绍Microsoft Office和OO.o脚本的编写，最后介绍一个Google Spreadsheet和Amazon.com Web服务的混搭。
- 第18章关于如何在网页中嵌入人和计算机程序都能理解的信息，提供了两个方案：微格式和RDFa。接着介绍如何使用和编写Firefox Operator扩展以识别和处理微格式。
- 第19章介绍如何使用Google Ajax Search API、Yahoo! Search API和Microsoft Live.com搜索。同时介绍OpenSearch和Google Desktop HTTP/XML网关。

目标读者

本书适用的读者群非常广泛，对于那些渴望创建Web 2.0应用或了解其背后支撑技术的人来说，本书都是非常合适的。作为阅读的基础，读者需要熟悉HTML，了解基本的CSS和JavaScript。我会提供一些相关背景材料的阅读指南。本书中大多数的服务器端代码都是基于PHP的，也有一部分用到了Python。

对于有经验的开发者而言，本书同样是非常有价值的。虽然涉及的内容相当广泛，但还是力图从混编的角度出发阐述一些对当前技术来说比较深层和本质的问题——这些东西常常并不是那么显而易见。

学习信息混编很容易陷入各种技术使用混乱的境地。为了理解和数据混编相关的XML、Web服务、Ajax、COM和元数据等内容，初学者常常需要付出艰苦的努力。其实混搭的起步并不是那么困难，只是需要你对很多不同的内容有所了解，此外学习的兴趣和一点冒险精神也必不可少。通常情况下，这些主题都散布在一大堆同类图书中，本书将指导你从何处开始。

关于修订

请参见<http://mashupguide.net>，其中包含内容更新以及和本书相关的一些补充材料。

致谢

我深深感谢那些为本书手稿提供了详细反馈信息的读者，他们是Jason Cooper、Andres Ferrate、Michael Kaply、John Musser、Paul Rademacher、Jon Udell和C. K. Yuan。John Musser无私地提供了访问Programmableweb.com上的数据的途径。

如果没有以前在加州大学伯克利分校的同事们的启发，我可能根本无法写成本书。我想特别感谢

Chris Ashley、Isaac Mankita和Susan Stone，他们在午饭或茶歇时耐心地倾听我滔滔不绝地谈论本书，这帮助我坚持到了最后。Thom King和Tom Schirmer慷慨地与我分享了他们关于数据架构和软件开发的知識。Shifra Gaman和Rich Meyer提供了许多技术支持和精神鼓励。感谢David Greenbaum对我教学的支持。感谢Rick Jaffe，一个参与我课程的同事，他一直在不懈地支持我的教学。Sara Leavitt和Aron Roberts无私地帮助我学习iCalendar和了解Berkeley Events Calendar。

加州大学伯克利分校信息学院是我这几年来一个重要的知识宝库。我非常感激能够获得教授“Mixing and Remixing Information”（信息的混合与混编）这门课程的机会，也很庆幸能和这里所有的人共事。最重要的，我要感谢我的学生，他们的关注以及所创建的精彩项目给了我很大的启发。同样感谢我的那些十几岁的夏令营学生，他们在短短六周的时间里创建了自己的混搭项目，并教会我什么是音乐混搭。

感谢Apress出版社的同仁，他们使得本书更加完美。除了不断地提出中肯的编辑建议，Matt Moodie在我写作的最后几个月给了我充分的鼓励。Richard Dal Parto提供了良好的项目管理，而Grace Wong则使之变得更加完美。我还要感谢生产编辑Laura Esterman和Lori Bring、高级制作编辑Kelly Winkuisth以及文字编辑Kim Wimpsett，是他们把代码和文本变成了符合出版要求的格式。我要感谢Tina Nielsen，她是我在Apress所接触的第一个人。特别感谢Chris Mills，作为我的第一位编辑，她的热情和所提供的详细反馈信息，为本书的编写工作开了一个好头。

我从John Watson精辟犀利的评论中学到了很多，他是本书的技术审稿人，同时也是非常了不起的Flickr混搭的作者！

我要感谢我的朋友们，他们激励了我，也为我庆祝。我的家庭总是在背后支持着我。谢谢他们的爱，感谢他们在我接受这个宏大的项目时对我的信任。

最后，我要感谢我的太太Laura，感谢在编写本书期间她为我所做的每一件事：她的爱、鼓励和充满智慧的建议；聆听我的每个刚刚萌生的想法；对书稿进行编辑以及为了让我有充足的时间写书而做出的牺牲。现在，是时候开始我们已推迟很久的假期了！

目 录

第一部分 混编信息，无需编程

第 1 章 从具体的混搭中学习 2

- 1.1 寻找混搭的模式 2
- 1.2 Housingmaps.com 3
 - 1.2.1 组合了什么 4
 - 1.2.2 为什么进行组合？解决了什么问题 4
 - 1.2.3 在哪里进行混编 5
 - 1.2.4 如何进行组合 5
 - 1.2.5 可供对比的混搭 6
- 1.3 Flickr 中的 Google Maps 6
 - 1.3.1 组合了什么 6
 - 1.3.2 为什么进行组合？解决了什么问题 7
 - 1.3.3 如何进行组合 10
 - 1.3.4 可供对比的混搭 10
- 1.4 LibraryLookup 书签小工具 11
 - 1.4.1 配置一个 LibraryLookup 书签小工具 11
 - 1.4.2 启动 LibraryLookup 书签小工具 12
 - 1.4.3 该混搭如何工作 13
 - 1.4.4 如何扩展该混搭 13
 - 1.4.5 可供对比的混搭 14
- 1.5 跟踪其他的混搭 14
- 1.6 小结 14

第 2 章 发现 Web 站点混搭的潜质 16

- 2.1 是什么使 Web 站点和应用能被混搭 16
 - 2.1.1 Web 站点的实体构成 17
 - 2.1.2 公开 API 和现有的混搭 18
 - 2.1.3 Ajax 的应用 18

- 2.1.4 嵌入脚本的能力 19
- 2.1.5 浏览器插件 19
- 2.1.6 Web 站点的数据的导入与导出 19
- 2.1.7 用户与开发者社区 20
- 2.1.8 移动界面、可替换界面以及站点的皮肤的定制性 20
- 2.1.9 文档 20
- 2.1.10 Web 站点是否开源 21
- 2.1.11 标签、提要和博客 21
- 2.2 Web 站点的 URL 语言 21
- 2.3 已知混搭的简要回顾 22
 - 2.3.1 最基本的对混搭友好的 Flickr 22
 - 2.3.2 Flickr 中的资源 23
 - 2.3.3 用户与照片 24
 - 2.3.4 照片的表示 24
 - 2.3.5 单张照片相关的数据 26
 - 2.3.6 标签 27
 - 2.3.7 用户图档：按日期浏览照片 28
 - 2.3.8 集 29
 - 2.3.9 集合 29
 - 2.3.10 收藏 30
 - 2.3.11 用户的流行照片 30
 - 2.3.12 联系人 30
 - 2.3.13 组 30
 - 2.3.14 账户管理 32
 - 2.3.15 走遍 Flickr 32
 - 2.3.16 搜索 33
 - 2.3.17 Flickr 中带地理标签的照片 34
 - 2.3.18 Flickr 管理工具 35
 - 2.3.19 近期活动 35
 - 2.3.20 邮件界面 35
 - 2.3.21 博客界面 36

2.3.22	提要综合: RSS 和 Atom	36	3.3	在 Technorati 中基于标签收集内容	59
2.3.23	移动访问	36	3.3.1	在 Technorati 中利用标签进行 搜索	59
2.3.24	第三方 Flickr 应用	36	3.3.2	Technorati 如何找到 Web 上的 标签	59
2.3.25	知识共享许可	37	3.3.3	Technorati 标签中的单词变形与 句法限制	59
2.3.26	相机	37	3.4	使用标签混搭 Flickr 和 del.icio.us	60
2.4	基于 URL 模板和 URL 嵌入的混搭 模式	37	3.5	其他使用标签的系统	60
2.5	Google Maps	39	3.6	标签与正规分类模式的关系	61
2.5.1	Google Maps 的 URL 语言	40	3.7	小结	62
2.5.2	在 Google Maps 中浏览 KML 文件	41	第 4 章	使用提要、RSS 和 Atom	63
2.5.3	连接 Yahoo! Pipes 和 Google Maps	41	4.1	什么是提要, 它为何如此重要	63
2.5.4	Google Maps URL 语言的 其他应用	42	4.2	RSS 2.0	64
2.6	Amazon	42	4.3	RSS 1.0	65
2.6.1	Amazon 中的商品	43	4.4	Atom 1.0	67
2.6.2	列表	44	4.5	RSS 2.0 和 Atom 1.0 的扩展	69
2.6.3	标签	45	4.6	来自 Flickr 的提要	70
2.6.4	主题的指向	45	4.6.1	Flickr 提要的参数	70
2.7	del.icio.us	46	4.6.2	Flickr 提要分析	71
2.8	屏幕抓取和机器人	47	4.6.3	RSS 和 Atom 之外的数据交换 格式	74
2.9	小结	48	4.7	来自其他 Web 站点的提要	75
第 3 章	理解标签和分众标签	50	4.7.1	找到提要与提要的自动发现	76
3.1	Flickr 的标签机制	51	4.7.2	博客提要	77
3.1.1	Flickr 中的标签	51	4.7.3	维基百科提要	77
3.1.2	标签的实际使用	52	4.7.4	Google News 和 Yahoo! News	78
3.1.3	创建自己的标签	53	4.8	新闻聚合器: 在别处显示 Flickr 提要	79
3.1.4	Flickr 的标签语法	53	4.9	提要的验证	80
3.1.5	标签的潜在弱点	53	4.10	使用 GUI 工具抓取提要	80
3.1.6	Flickr 标签的单/复数形式	54	4.11	使用 Feedburner 混编提要	81
3.1.7	切入标签系统: 地理标签与 机器标签	54	4.12	使用 Yahoo! Pipes 混编提要	82
3.1.8	Flickr 标签的有趣应用	55	4.12.1	Yahoo! News: 第一个简单 管道	82
3.2	del.icio.us 的标签机制	56	4.12.2	Google News 与管道的重构	83
3.2.1	在 del.icio.us 中添加标签的 技巧	56	4.12.3	Wikinews 和 NY Times: 提要过滤	84
3.2.2	大小写和多词短语的处理	56	4.12.4	提要的组合	84
3.2.3	获得更多信息	57	4.13	小结	85

第 5 章 博客的集成	86
5.1 博客集成的场景	86
5.2 将 Flickr 照片发送到博客	87
5.2.1 配置 Flickr 与博客的集成	88
5.2.2 在博客上发布 Flickr 照片	90
5.2.3 Flickr 的博客集成机制	90
5.3 桌面博客发布工具	91
5.4 组合提要与博客发布工具来生成 反馈流	92
5.5 Flock: 博客与 Flickr 的结合	93
5.6 RSD: 博客 API 的可发现性	94
5.7 反向链接	95
5.8 初级 Wiki 集成	96
5.9 小结	97

第二部分 使用 API 混编单个 Web 应用

第 6 章 通过 Flickr 学习 Web 服务 API	100
6.1 Flickr API 简介	101
6.1.1 XML 响应的含义	103
6.1.2 能对 XML 响应做些什么	104
6.2 API 文档、社群与策略	105
6.3 使用 Flickr API Explorer 及其文档	106
6.4 使用 PHP 调用基本的 Flickr API 方法	109
6.4.1 HTTP 客户端	110
6.4.2 关于 HTTP	111
6.4.3 处理 XML	114
6.4.4 整合所有信息: 为照片演示生成 一个简单的 HTML	118
6.4.5 启示	120
6.5 一般的 Flickr API	120
6.5.1 使用 flickr.reflection 方法	121
6.5.2 使用 PHP 查询 Flickr 反射方法	123
6.6 请求与响应格式	128
6.7 Flickr 授权	130
6.7.1 为什么不应该分发用户口令	130
6.7.2 Web 应用授权	131
6.8 使用 Flickr API 工具包	137
6.8.1 PEAR::Flickr_API	138

6.8.2 phpFlickr	139
6.8.3 Phlickr	140
6.9 Flickr API 的局限性	141
6.10 小结	142
第 7 章 其他 Web API	143
7.1 XML-RPC	144
7.1.1 传输中发生了什么	148
7.1.2 使用 Wireshark 和 curl 进行 HTTP 消息分析与构造	148
7.1.3 解析 XML-RPC 数据流	149
7.2 SOAP	152
7.2.1 梦想: 通过 WSDL 和 SOAP 实现即插即用式功能	152
7.2.2 geocoder.us	153
7.2.3 Amazon ECS	160
7.2.4 通过 SOAP 使用 Flickr API	163
7.3 学习具体的 Web API	163
7.3.1 Programmableweb.com	164
7.3.2 YouTube	166
7.3.3 GData 与 Blogger API	167
7.3.4 将 Blogger API 作为基于 HTTP 方法的统一接口使用	170
7.4 小结	170
第 8 章 Ajax/JavaScript 微件及其 API	172
8.1 必要的知识	173
8.2 Ajax 带来了什么	173
8.3 Firebug, DOM Inspector 和 JavaScript Shell	174
8.3.1 使用 DOM Inspector	174
8.3.2 使用 Firefox 的 Firebug 扩展	174
8.3.3 使用 JavaScript Shell	175
8.4 使用 JavaScript 库	176
8.5 YUI 微件	176
8.5.1 使用 YUI Calendar	177
8.5.2 安装 YUI	178
8.6 了解 Google Maps	178
8.7 使用 JavaScript 访问 Flickr	181
8.8 使用 Greasemonkey 访问《纽约时报》	

的永久链接	184
8.9 更多地了解 JavaScript 和 Ajax	187
8.10 小结	187
 第三部分 创建混搭	
第 9 章 从 API 和可混编元素到混搭	190
9.1 ProgrammableWeb	190
9.1.1 ProgrammableWeb 中由用户 提供的数据	191
9.1.2 混搭的所有目录都一直存在吗	191
9.1.3 了解混搭的全貌	191
9.1.4 混搭的目录	192
9.1.5 使用提要跟踪混搭	193
9.1.6 使用标签描述混搭	193
9.1.7 API 与混搭的垂直市场	195
9.2 查看特定的混搭概要信息	195
9.3 从特定的 API 到混搭	196
9.4 示例: 准备用混搭解决的问题	196
9.4.1 跟踪感兴趣的图书	196
9.4.2 何时购买机票	200
9.4.3 寻找梦幻家园	201
9.4.4 将发生的新闻绘制在地图上	201
9.5 小结	202
第 10 章 创建多个服务的混搭	203
10.1 设计	204
10.2 背景回顾: Flickr 中的地理标签	204
10.3 背景回顾: XMLHttpRequest 及其所 包含的库	207
10.3.1 直接使用 XMLHttpRequest	207
10.3.2 使用 YUI Connection Manager	209
10.4 建立一个服务器端代理	211
10.4.1 在 XHR 和直接的 API 调用中 都发生了什么	212
10.4.2 为带有地理信息的照片创建 一个服务器端脚本	213
10.5 建立一个简单的客户端框架	215
10.5.1 读写各种元素	215
10.5.2 处理简单事件以连接表单输入 并显示计算结果	217
10.6 将客户端框架与 Flickr 挂接	218
10.6.1 编写向 flickrgeo.php 查询的 URL	219
10.6.2 通过 YUI Connection Manager 使用 XHR 读入 JSON	219
10.6.3 将 JSON 转换为 HTML	221
10.7 混搭 Google Maps API 和 Flickr	223
10.7.1 建立基本的 Google 地图	223
10.7.2 使地图对地图视口的改变 做出响应	224
10.7.3 将 Flickr 和 GMap 代码放在 一起	226
10.7.4 获取 Google 地图的范围框	226
10.7.5 把照片显示在地图上	228
10.8 显示 Flickr 照片的 Google Mapplet	232
10.9 小结	236
第 11 章 使用工具创建混搭	237
11.1 混搭工具所解决的问题	237
11.2 本章将创建的混搭	238
11.3 一步步生成混搭	239
11.3.1 熟悉 Google Mashup Editor	240
11.3.2 读入并显示一个提要 (简单模板)	240
11.3.3 插入用户模板	241
11.3.4 使用 Yahoo! Pipes 访问 Flickr	243
11.3.5 使用 <gm:map> 显示 Flickr 照片	244
11.3.6 为混搭添加 JavaScript	246
11.3.7 使用选项卡持久保存提要	250
11.3.8 最终产品: 在地图上显示 已保存的条目	254
11.4 关于 GME 和 Yahoo! Pipes 的权衡	259
11.5 其他混搭工具	260
11.6 小结	261
第 12 章 使你的 Web 站点可混搭	262
12.1 为什么要使 Web 站点可混搭	262

12.2 不依赖于 API 的技术手段	263	12.3.17 帮助 API 用户更好地使用 资源	271
12.2.1 使用具有持久性和含义 丰富的 URL 语言	263	12.3.18 开放应用代码	271
12.2.2 使用 W3C 标准进行网站 开发	263	12.4 易于理解的数据标准	271
12.2.3 Web 的可达性	264	12.5 小结	272
12.2.4 允许用户为内容添加标签	264		
12.2.5 使提要可用	264	第四部分 混搭的其他主题	
12.2.6 向博客和其他 Web 站点 轻松发布内容	265	第 13 章 混编在线地图与 3D 数字化 地球	274
12.2.7 鼓励带有显式许可的内容 共享	265	13.1 在线地图的数量	275
12.2.8 为用户内容提供导入导出的 扩展选项	266	13.2 基于地图的混搭示例	275
12.2.9 研究用户如何混编你的内容 并使之更加容易	266	13.3 无需编程制作地图	276
12.3 创建对混搭友好的 API	266	13.3.1 Mapbuilder.net	276
12.3.1 学习和模仿其他 API	266	13.3.2 Google My Maps	277
12.3.2 从观众的角度出发	267	13.3.3 利用混搭的机会: Yahoo! Local 集合的地图表示	278
12.3.3 创建易于了解的 API	267	13.3.4 将 Yahoo! Local 的 XML 转换 为 CSV 供 Mapbuilder.net 使用	280
12.3.4 测试 API 的可用性	267	13.3.5 在微软的 Live Search Maps 中 创建集合	282
12.3.5 建立细粒度、松耦合的体系, 使 API 能够向所有人提供 相同服务	268	13.3.6 无需编程创建地图小结	283
12.3.6 拥抱 REST, 但也尽可能支持 SOAP 和 XML-RPC	268	13.4 数据交换格式	283
12.3.7 使用 APP 作为 REST 的一个 具体实例	268	13.4.1 CSV	284
12.3.8 鼓励开发第三方或内部的 API 工具包	268	13.4.2 HTML 微格式与元标签	284
12.3.9 在 API 中支持可扩展的错误 报告机制	269	13.4.3 GeoRSS	285
12.3.10 接受多种输入输出格式	269	13.4.4 Yahoo!对 GeoRSS 的使用与 Yahoo! YMaps 扩展	286
12.3.11 在 API 中支持 UI 功能	269	13.4.5 KML	290
12.3.12 提供搜索 API	270	13.4.6 格式互操作: GeoRSS 与 KML	290
12.3.13 API 版本处理	270	13.5 通过 API 编程创建地图	291
12.3.14 培育开发者社群	270	13.5.1 Google Maps API	291
12.3.15 不要对 API 过度控制	270	13.5.2 Yahoo! Maps API	294
12.3.16 生产服务水平协议 (SLA)	270	13.5.3 微软的 Live Search Maps/Virtual Earth	297
		13.6 地理编码	299
		13.6.1 Yahoo! Maps	299
		13.6.2 Geocoder.us	300

13.6.3	Google Geocoder.....	301	15.2.1	最终用户指南.....	368
13.6.4	Virtual Earth.....	303	15.2.2	30boxes.com API.....	369
13.6.5	非美国境内地址的地理编码 ...	305	15.3	事件聚合器.....	371
13.7	Google Earth 与 KML.....	305	15.3.1	Upcoming.yahoo.com.....	372
13.7.1	以最终用户身份显示和处理 KML	306	15.3.2	Eventful.com	379
13.7.2	KML	309	15.4	对 iCalendar 编程	385
13.7.3	通过 COM 和 AppleScript 对 Google Earth 编程	314	15.4.1	Python 与 iCalendar	385
13.8	Mapstraction 与 OpenLayers.....	315	15.4.2	PHP 与 iCalendar	386
13.9	综合示例: 在 Google Earth 中显示 Flickr 图片	316	15.5	向 iCalendar 和 Google Calendar 导出 一个事件日历.....	387
13.9.1	KML NetworkLink.....	318	15.5.1	信息源: UC Berkeley Event Calendars	388
13.9.2	为照片生成 KML	320	15.5.2	使用 Python 创建 Critic Choice 的 iCalendar 提要.....	388
13.9.3	flickrgeo.php 源代码	322	15.5.3	将事件写入 Google Calendar ...	390
13.10	小结.....	331	15.6	小结.....	396
第 14 章	探索社会化书签与目录系统.....	332	第 16 章	使用在线存储服务.....	398
14.1	社会化书签发展场景	333	16.1	Amazon S3 简介	398
14.2	del.icio.us	334	16.2	S3 的基本原理.....	399
14.2.1	使用 del.icio.us API	335	16.3	Amazon S3 的概念结构	399
14.2.2	del.icio.us 的第三方工具.....	340	16.4	Firefox 的 S3 扩展: 初步使用 S3	401
14.2.3	第三方 API 工具包.....	340	16.5	使用 S3 的 REST 接口	401
14.3	Yahoo! Bookmarks 与 Yahoo! MyWeb.....	342	16.6	使用 S3 的 SOAP 接口.....	405
14.4	Connotea.....	343	16.7	Amazon S3 的 API 工具包.....	406
14.5	Flickr 与 del.icio.us 的混搭	346	16.7.1	PHP	406
14.6	小结.....	349	16.7.2	Python	407
第 15 章	访问在线日历与事件聚合器.....	350	16.8	小结.....	408
15.1	Google Calendar.....	351	第 17 章	混搭桌面和基于 Web 的 办公套件	409
15.1.1	作为最终用户建立 Google Calendar	351	17.1	办公套件的混搭场景	409
15.1.2	Google Calendar 的提要格式 ...	353	17.2	文档的混搭	410
15.1.3	直接使用基于 GData 的 Calendar API.....	358	17.3	OpenDocument 格式.....	410
15.1.4	使用 Google Calendar 的 PHP API 工具包.....	364	17.3.1	基本 ODF 标签	418
15.1.5	使用 Google Calendar 的 Python API 工具包.....	367	17.3.2	创建不含任何样式元素的 ODF 文本文档	419
15.2	30boxes.com.....	368	17.3.3	将段落文本设为 text-body.....	423
			17.3.4	格式化列表以区分有序号和 无序号列表	423

17.3.5 在 Text Span 中设置黑体、 斜体、字型和颜色	425	18.3.1 rel-design-pattern	460
17.4 ODF 的 API 工具包	426	18.3.2 class-design-pattern	460
17.4.1 Odfpy	426	18.3.3 abbr-design-pattern	460
17.4.2 OpenDocumentPHP	434	18.3.4 include-pattern	461
17.4.3 使用 OO.o 生成 ODF	437	18.4 各种微格式	461
17.5 ECMA Office Open XML (OOXML)	437	18.4.1 rel-license	462
17.6 比较 ODF 和 OOXML	440	18.4.2 rel-tag	462
17.7 在线办公套件	440	18.4.3 xfn	462
17.8 Google Spreadsheets API	441	18.4.4 xFolk	463
17.8.1 Python API 工具包	441	18.4.5 geo	463
17.8.2 混搭: Amazon 期望表与 Google Spreadsheets	445	18.4.6 hCard 与 adr	464
17.8.3 Google Spreadsheets 的 Zend PHP API 工具包	449	18.4.7 hCalendar	464
17.8.4 最后的变化: 通过 COM 将 Amazon 期望表转换为 Microsoft Excel	451	18.4.8 其他微格式	465
17.9 Zoho API	452	18.5 实际应用中的微格式	465
17.10 小结	452	18.6 对微格式编程	465
第 18 章 可嵌入数据格式: 微格式 和 RDFa	453	18.7 编写 Operator 脚本	466
18.1 使用 Operator 学习微格式	453	18.7.1 研究指导性的脚本	467
18.1.1 adr (地址)	456	18.7.2 编写地理编码脚本	468
18.1.2 hCard (联系方式)	456	18.8 资源 (RDFa): 有望成为微格式的 补充	470
18.1.3 hCalendar (事件)	457	18.9 进一步学习的参考资料	470
18.1.4 geo (位置)	458	18.10 小结	471
18.1.5 tag (tag space)	458	第 19 章 搜索的集成	472
18.2 微格式的定义及其设计目标	458	19.1 Google Ajax Search	472
18.3 微格式的设计模式	459	19.2 Yahoo! Search	474
		19.3 Microsoft Live.com Search	476
		19.4 OpenSearch	480
		19.5 Google Desktop HTTP/XML 网关	482
		19.6 小结	483
		附录 知识产权、可重用性与知识共享	484