

Pro Web 2.0 Mashups

Remixing Data and Web Services



Raymond Yee

Pro Web 2.0 Mashups: Remixing Data and Web Services

Copyright © 2008 by Raymond Yee

Permission is granted to copy, distribute, and/or modify this document under the terms of the Creative Commons Attribution–NonCommercial–ShareAlike–2.5 License. Apress (<http://www.apress.com/>) and the author ask for your support by buying the print or eBook edition through any online or retail outlet. A copy of the license is included in the section entitled “Creative Commons Legal Code.” All rights reserved subject to the provisions of the Creative Commons License.

ISBN-13 (pbk): 978-1-59059-858-0

ISBN-10 (pbk): 1-59059-858-X

ISBN-13 (electronic): 978-1-4302-0286-8

ISBN-10 (electronic): 1-4302-0286-6

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Matthew Moodie

Technical Reviewer: John Watson

Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell,

Jonathan Gennick, Kevin Goff, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann,

Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Richard Dal Porto

Copy Editor: Kim Wimpsett

Associate Production Director: Kari Brooks-Copony

Production Editors: Laura Esterman, Lori Bring

Compositor: Kinetic Publishing Service, LLC

Proofreader: Liz Welch

Indexer: Broccoli Information Management

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at <http://www.apress.com/info/bulksales>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com>. You may need to answer questions pertaining to this book in order to successfully download the code.

For Laura, the love of my life

Contents at a Glance

About the Author	xxi
About the Technical Reviewer	xxiii
Acknowledgments	xxv
Introduction	xxvii

PART 1 ■ ■ ■ Remixing Information Without Programming

■ CHAPTER 1	Learning from Specific Mashups	3
■ CHAPTER 2	Uncovering the Mashup Potential of Web Sites	21
■ CHAPTER 3	Understanding Tagging and Folksonomies	61
■ CHAPTER 4	Working with Feeds, RSS, and Atom	77
■ CHAPTER 5	Integrating with Blogs	105

PART 2 ■ ■ ■ Remixing a Single Web Application Using Its API

■ CHAPTER 6	Learning Web Services APIs Through Flickr	121
■ CHAPTER 7	Exploring Other Web APIs	171
■ CHAPTER 8	Learning Ajax/JavaScript Widgets and Their APIs	205

PART 3 ■ ■ ■ Making Mashups

■ CHAPTER 9	Moving from APIs and Remixable Elements to Mashups	227
■ CHAPTER 10	Creating Mashups of Several Services	243
■ CHAPTER 11	Using Tools to Create Mashups	283
■ CHAPTER 12	Making Your Web Site Mashable	313

PART 4 ■ ■ ■ Exploring Other Mashup Topics

■ CHAPTER 13	Remixing Online Maps and 3D Digital Globes	327
■ CHAPTER 14	Exploring Social Bookmarking and Bibliographic Systems	395
■ CHAPTER 15	Accessing Online Calendars and Event Aggregators	417
■ CHAPTER 16	Using Online Storage Services	473
■ CHAPTER 17	Mashing Up Desktop and Web-Based Office Suites	487
■ CHAPTER 18	Using Microformats and RDFa As Embeddable Data Formats	537
■ CHAPTER 19	Integrating Search	559
■ APPENDIX	573
■ INDEX	579

Contents

About the Author	xxi
About the Technical Reviewer	xxiii
Acknowledgments	xxv
Introduction	xxvii

PART 1 ■ ■ ■ **Remixing Information Without Programming**

■ CHAPTER 1	Learning from Specific Mashups	3
	Looking for Patterns in Mashups	3
	Housingmaps.com	5
	What Is Being Combined?	5
	Why Are the Constituent Elements Being Combined? What's the Problem Being Solved?	5
	Where Is the Remixing Happening?	6
	How Are These Elements Being Combined?	6
	Comparable Mashups	7
	Google Maps in Flickr	7
	What Is Being Combined?	8
	Why Are the Constituent Elements Being Combined? What's the Problem Being Solved?	8
	How Are These Elements Being Combined?	12
	Comparable Mashups	13
	LibraryLookup Bookmarklet	13
	Configuring a LibraryLookup Bookmarklet	14
	Invoking the LibraryLookup Bookmarklet	15
	How Does This Mashup Work?	16
	How Can This Mashup Be Extended?	17
	Comparable Mashups	18
	Tracking Other Mashups	18
	Summary	18

CHAPTER 2	Uncovering the Mashup Potential of Web Sites	21
What Makes Web Sites and Applications Mashable		22
Ascertaining the Fundamental Entities of the Web Site		22
Public APIs and Existing Mashups		23
Use of Ajax		24
Embedded Scriptability		24
Browser Plug-Ins		25
Getting Data In and Out of the Web Site		25
The Community of Users and Developers		25
Mobile and Alternative Interfaces and the Skinnability of the Web Site		26
Documentation		26
Is the Web Site Run on Open Source?		26
Intellectual Property, Reusability, and Creative Commons		26
Tagging, Feeds, and Weblogging		27
URL Languages of Web Sites		27
Some Mashups Briefly Revisited		28
Flickr: The Fundamentally Mashup-Friendly Site		29
Resources in Flickr		29
Users and Photos		30
Data Associated with an Individual Photo		33
Tags		34
User's Archive: Browsing Photos by Date		36
Sets		37
Collections		37
Favorites		37
A User's Popular Photos		38
Contacts		38
Groups		38
Account Management		40
Browsing Through Flickr		40
Search		41
Geotagged Photos in Flickr		42
The Flickr Organizer		43
Recent Activities		44
Mailing Interfaces		44
Interfacing to Weblogs		44
Syndication Feeds: RSS and Atom		45
Mobile Access		45
Third-Party Flickr Apps		45

Creative Commons Licensing	46
Cameras	46
The Mashup-by-URL-Templating-and-Embedding Pattern	47
Google Maps	49
URL Language of Google Maps	49
Viewing KML Files in Google Maps	51
Connecting Yahoo! Pipes and Google Maps	51
Other Simple Applications of the Google Maps URL Language	52
Amazon	53
Amazon Items	53
Lists	55
Tags	55
Subject Headings	55
del.icio.us	56
Screen-Scraping and Bots	58
Summary	60

■ CHAPTER 3 **Understanding Tagging and Folksonomies** 61

Tagging in Flickr	62
Tags in Flickr	63
How Tags Are Used in Practice	63
Creating Your Own Tags	64
Syntax of Tags in Flickr	64
Potential Weaknesses of Tags	65
Singular and Plural Forms of Tags in Flickr	65
Hacking the Tagging System: Geotagging and Machine Tags	66
Interesting Apps Using Flickr Tags	67
Tagging in del.icio.us	67
Mechanics of Adding Tags in del.icio.us	68
Dealing with Case and Multiword Phrases	68
Getting More Information	69
Gathering Content Through Tags in Technorati	71
Searching Technorati with Tags	71
How Technorati Finds Tags on the Web	72
Word Inflections and Syntactic Constraints in Technorati Tags	72
Using Tags to Mash Up Flickr and del.icio.us	72
Other Systems That Use Tagging	73
Relationship of Tags to Formal Classification Schemes	73
Summary	75

CHAPTER 4	Working with Feeds, RSS, and Atom	77
	What Are Feeds, and Why Are They Important?	78
	RSS 2.0	78
	RSS 1.0	80
	Atom 1.0	82
	Extensions to RSS 2.0 and Atom 1.0	84
	Feeds from Flickr	86
	Flickr Feed Parameters	86
	Examining the Flickr Feeds	87
	Exchange Formats Other Than RSS and Atom	90
	Feeds from Other Web Sites	92
	Finding Feeds and Feed Autodiscovery	93
	Feeds from Weblogs	94
	Wikipedia Feeds	94
	Google and Yahoo! News	95
	News Aggregators: Showing Flickr Feeds Elsewhere	96
	Validating Feeds	98
	Scraping Feeds Using GUI Tools	98
	Remixing Feeds with Feedburner	99
	Remixing Feeds with Yahoo! Pipes	100
	A Simple First Pipe with Yahoo! News	101
	Google News and Refactoring Pipes	102
	Wikinews and NY Times: Filtering Feeds	103
	Pulling the Feeds Together	104
	Summary	104
CHAPTER 5	Integrating with Blogs	105
	Integration Scenarios for Blogs	105
	Sending Flickr Pictures to Blogs	106
	Configuring Flickr for Integration with Blogs	107
	Blogging a Flickr Picture	110
	How Does the Flickr Blog Integration Work?	110
	Desktop Blogging Tools	111
	Combining Feeds and Blogging to Generate Feedback Flows	113
	Flock: Bringing Together Blogs and Flickr	114
	RSD: Discoverability of Blog APIs	115
	Linkbacks	116
	Wiki Integration at an Early Stage	116
	Summary	117

PART 2 ■ ■ ■ Remixing a Single Web Application Using Its API

CHAPTER 6	Learning Web Services APIs Through Flickr	121
	An Introduction to the Flickr API	122
	What Does This XML Response Mean?	124
	What Can You Do with the XML Response?	126
	API Documentation, Community, and Policy	128
	Terms of Use for the API	128
	Using the Flickr API Explorer and Documentation	129
	Calling a Basic Flickr API Method from PHP	132
	HTTP Clients	133
	A Refresher on HTTP	134
	XML Processing	138
	Pulling It All Together: Generating Simple HTML Representations of the Photos	143
	Where Does This Leave Us?	145
	The Flickr API in General	145
	Using flickr.reflection Methods	146
	Querying the Flickr Reflection Methods with PHP	149
	Request and Response Formats	154
	Flickr Authorization	156
	Why Passing Passwords Around Doesn't Work Too Well	157
	Authorization for Web Apps	157
	Using Flickr API Kits	165
	PEAR::Flickr_API	165
	phpFlickr	166
	Phlickr	168
	Limitations of the Flickr API	169
	Summary	170
CHAPTER 7	Exploring Other Web APIs	171
	XML-RPC	172
	What's Happening on the Wire?	176
	Using Wireshark and curl to Analyze and Formulate HTTP Messages	177
	Parsing XML-RPC Traffic	178

SOAP	181
The Dream: Plug-and-Go Functionality Through WSDL and SOAP	181
geocoder.us	182
Amazon ECS	191
The Flickr API via SOAP	195
Learning About Specific Web APIs	195
Programmableweb.com	196
YouTube	198
GData and the Blogger API	199
Using the Blogger API As a Uniform Interface Based on HTTP Methods	203
Summary	204

■ CHAPTER 8 **Learning Ajax/JavaScript Widgets and Their APIs** 205

What You Need to Know	206
What Difference Does Ajax Make?	207
Learning Firebug, DOM Inspector, and JavaScript Shell	208
Using the DOM Inspector	208
Using the Firebug Extension for Firefox	208
Using the JavaScript Shell	210
Working with JavaScript Libraries	210
YUI Widgets	211
Using the YUI Calendar	211
Installing YUI on Your Host	212
Learning Google Maps	213
Accessing Flickr via JavaScript	217
Using Greasemonkey to Access New York Times Permalinks	220
Learning More About JavaScript and Ajax	223
Summary	223

PART 3 ■ ■ ■ **Making Mashups**

■ CHAPTER 9 **Moving from APIs and Remixable Elements to Mashups** 227

Getting Oriented to ProgrammableWeb	228
User-Generated Data in ProgrammableWeb	228
Can Any Directory of Mashups Keep Up?	228
Learning About the Overall Mashup Scene	229

Directory of Mashups	230
Using Feeds to Track Mashups	230
Using Tags to Describe Mashups	231
API and Mashup Verticals	233
Looking at a Specific Mashup Profile	233
Going from a Specific API to Mashups	234
Sample Problems to Solve Using Mashups	235
Tracking Interesting Books	235
Knowing When to Buy Airplane Tickets	239
Finding That Dream House	240
Mapping Breaking News	241
Summary	242

■ CHAPTER 10 Creating Mashups of Several Services

The Design	244
Background: Geotagging in Flickr	245
Background: XMLHttpRequest and Containing Libraries	248
Using XMLHttpRequest Directly	248
Using the YUI Connection Manager	250
Building a Server-Side Proxy	253
What Happens with XHR and Direct API Calls?	253
Building a Server-Side Script for Geolocated Photos	255
Building a Simple Client-Side Frame	257
Reading and Writing Elements	257
Handling Simple Events to Connect Form Input and Display Calculations	260
Hooking the Client-Side Framework to Flickr	261
Writing a URL for Querying flickrgeo.php	262
Using XHR via the YUI Connection Manager to Read the JSON	262
Converting the JSON to HTML	264
Mashing Up Google Maps API with Flickr	266
Setting Up a Basic Google Map	267
Making the Map Respond to Changes in the Viewport of the Map	268
Bringing Together the Flickr and GMap Code	269
Wiring Up the Bounding Box of the Google Map	270
Making the Pictures Show Up in the Map	272
Google Maplet That Shows Flickr Photos	277
Summary	281

CHAPTER 11	Using Tools to Create Mashups	283
	The Problem Mashup Tools Solve	284
	What You Are Making in This Chapter	284
	Making the Mashup: A Step-by-Step Example	286
	Familiarizing Yourself with the Google Mashup Editor	287
	Reading and Displaying a Feed (Simple Template)	288
	Introducing a Custom Template	289
	Using Yahoo! Pipes to Access Flickr	291
	Displaying Flickr Photos Using <code><gm:map></code>	292
	Adding JavaScript to the Mashup	294
	How to Persist Feeds and Use Tabs	299
	The Final Product: Showing the Saved Entries on a Map	304
	Analysis of Trade-Offs in Using GME and Yahoo! Pipes	309
	Other Mashup Tools	310
	Summary	311
CHAPTER 12	Making Your Web Site Mashable	313
	Why Make Your Web Site Mashable?	314
	Using Techniques That Do Not Depend on APIs	314
	Use a Consistent and Rich URL Language	314
	Use W3C Standards to Develop Your Web Site	315
	Pay Attention to Web Accessibility	315
	Consider Allowing Users to Tag Your Content	315
	Make Feeds Available	315
	Make It Easy to Post Your Content to Blogs and Other Web Sites	316
	Encourage the Sharing of Content with Explicit Licenses	317
	Develop Extensive Import and Export Options for User Content	317
	Study How Users Remix Your Content and Make It Easier to Do So	317
	Creating a Mashup-Friendly API	317
	Learn From and Emulate Other APIs	318
	Keep in Mind Your Audiences for the API	318
	Make Your API Easy to Learn	318
	Test the Usability of Your API	319
	Build a Granular, Loosely Coupled Architecture So That Creating an API Serves You As Much As It Does Others	319
	Embrace REST But Also Support SOAP and XML-RPC If You Can	320

Consider Using the Atom Publishing Protocol As a Specific Instantiation of REST	320
Encourage the Development of API Kits: Third Party or In-House.	320
Support Extensive Error Reporting in Your APIs	321
Accept Multiple Formats for Output and Input	321
Support UI Functionality in the API	321
Include a Search API for Your Own Site	321
Version Your API	322
Foster a Community of Developers.	322
Don't Try to Be Too Controlling in Your API.	322
Consider Producing a Service-Level Agreement (SLA)	322
Help API Users Consume Your Resources Wisely	323
Consider Open Sourcing Your Application	323
Easy-to-Understand Data Standards	323
Summary	324

PART 4 ■ ■ ■ Exploring Other Mashup Topics

■ CHAPTER 13 Remixing Online Maps and 3D Digital Globes	327
The Number of Online Maps	328
Examples of Map-Based Mashups	329
Making Maps Without Programming	329
Mapbuilder.net	329
Google My Maps	331
A Mashup Opportunity: Mapping Yahoo! Local Collections	332
Transforming the Yahoo! Local XML into CSV for Mapbuilder.net	334
Collection Building in Microsoft's Live Search Maps	336
Summary of Making Maps Without Programming	338
Data Exchange Formats	338
CSV	338
Microformats and Metatags for HTML	338
GeoRSS	339
Yahoo!'s Use of GeoRSS and Yahoo! YMaps Extensions	341
KML	345
Interoperability Among Formats: GeoRSS vs. KML	346

Creating Maps by API Programming	346
Google Maps API	347
Yahoo! Maps API	351
Microsoft's Live Search Maps/Virtual Earth	354
Geocoding	356
Yahoo! Maps	356
Geocoder.us	357
Google Geocoder	358
Virtual Earth	361
Geocoding Non-U.S. Addresses	363
Google Earth and KML	364
Displaying and Handling KML As End Users	364
KML	368
Programming Google Earth via COM and AppleScript	374
Mapstraction and OpenLayers	376
An Integrative Example: Showing Flickr Pictures in Google Earth	376
KML NetworkLink	379
Generating the KML for the Photos	382
The flickrgeo.php Code	383
Summary	393

■ CHAPTER 14 Exploring Social Bookmarking and Bibliographic Systems

The Social Bookmarking Scene	396
Using Programmableweb.com to Examine the Popularity of APIs	396
del.icio.us	397
Using the del.icio.us API	398
Third-Party Tools for del.icio.us	405
Third-Party API Kits	405
Yahoo! Bookmarks and MyWeb	407
Connotea	408
A Flickr and del.icio.us Mashup	412
Summary	416

■ CHAPTER 15 Accessing Online Calendars and Event Aggregators

Google Calendar	418
Setting Up Google Calendar As an End User	418
Exploring the Feed Formats from Google Calendar	420

Using the GData-Based Calendar API Directly	426
Using the PHP API Kit for Google Calendar	434
Using the Python API Kit for Google Calendar	437
30boxes.com	438
An End User Tutorial	439
30boxes.com API	439
Event Aggregators	443
Upcoming.yahoo.com	443
Eventful.com	452
Programming with iCalendar	458
Python and iCalendar	458
PHP and iCalendar	460
Exporting an Events Calendar to iCalendar and Google Calendar	461
The Source: UC Berkeley Event Calendars	462
Creating an iCalendar Feed of Critic's Choice Using Python	462
Writing the Events to Google Calendar	464
Summary	471

■ CHAPTER 16 Using Online Storage Services

Introducing Amazon S3	473
Rationale for S3	474
Conceptual Structure of Amazon S3	475
The Firefox S3 Extension Gets You Started with S3	476
Using the S3 REST Interface	477
Listing Buckets Using the REST Interface	480
Using the SOAP Interface to S3	481
Amazon S3 API Kits	482
PHP	483
Python	484
Summary	486

■ CHAPTER 17 Mashing Up Desktop and Web-Based Office Suites

Mashup Scenarios for Office Suites	487
The World of Document Markup	488
The OpenDocument Format	488
Learning Basic ODF Tags	497
Create an ODF Text Document Without Any Styling of	
ODF Elements	499
Setting the Paragraph Text to text-body	503

Formatting Lists to Distinguish Between Ordered and Unordered Lists	504
Getting Bold, Italics, Font Changes, and Color Changes into Text Spans	505
API Kits for Working with ODF	507
Odfpy	507
OpenDocumentPHP	516
Leveraging OO.o to Generate ODF	518
ECMA Office Open XML (OOXML)	519
Viewers/Validators for OOXML	522
Comparing ODF and OOXML	522
Online Office Suites	523
Usage Scenarios for Programmable Online Spreadsheets	523
Google Spreadsheets API	524
Python API Kit	524
Mashup: Amazon Wishlist and Google Spreadsheets Mashup	528
Zend PHP API Kit for Google Spreadsheets	533
A Final Variation: Amazon Wishlist to Microsoft Excel via COM	535
Zoho APIs	536
Summary	536

■ CHAPTER 18 Using Microformats and RDFa As Embeddable Data Formats

Using Operator to Learn About Microformats	537
adr (Addresses)	540
hCard (Contacts)	541
hCalendar (Events)	542
geo (Locations)	543
tag (Tagspaces)	543
Definitions and Design Goals of Microformats	543
Microformats Design Patterns	545
rel-design-pattern	545
class-design-pattern	545
abbr-design-pattern	546
include-pattern	546
Examples of Microformats	547
rel-license	547
rel-tag	548
xfn	548

xFolk	549
geo	549
hCard and adr	550
hCalendar	551
Other Microformats	551
Microformats in Practice	552
Programming with Microformats	552
Language-Specific Libraries	552
Writing an Operator Script	553
Studying the Tutorial Script	554
Writing a Geocoding Script	556
Resources (RDFa): A Promising Complement to Microformats	557
Reference for Further Study	558
Summary	558
 CHAPTER 19 Integrating Search	559
Google Ajax Search	559
Manipulating Search Results	559
Yahoo! Search	561
Yahoo! Images	563
Microsoft Live.com Search	564
OpenSearch	568
Google Desktop HTTP/XML Gateway	570
Summary	571
 APPENDIX	573
 INDEX	579

About the Author



■ **RAYMOND YEE** is a data architect, consultant, and trainer. He is currently a lecturer at the School of Information, UC Berkeley, where he teaches the course “Mixing and Remixing Information.” While earning a PhD in biophysics, he taught computer science, philosophy, and personal development to K–11 students in the Academic Talent Development Program on the Berkeley campus. He is the primary architect of the Scholar’s Box, software that enables users to gather digital content from multiple sources to create personal collections that can be shared with others. As a software architect and developer, he focuses on developing software to support learning, teaching, scholarship, and research.

Raymond is an erstwhile tubaist, admirer of J. S. Bach, Presbyterian elder, aspiring essayist, son of industrious Chinese-Canadian restaurateurs, and devoted husband of the incomparable Laura.

About the Technical Reviewer

■ **JOHN WATSON** is a professional freelance software developer and has been creating web-based software since 1994. He is best known on Flickr for fd's Flickr Toys (Bighugelabs.com), a popular collection of free photo manipulation utilities that use various APIs from Flickr, Google, and Yahoo! John is married and living happily in southern California with his wife and two young children. You can find out more about John and his recent projects at <http://watson-net.com>.

Acknowledgments

I'm deeply thankful to those who provided detailed feedback to parts of my manuscript: Jason Cooper, Andres Ferrate, Michael Kaply, John Musser, Paul Rademacher, Jon Udell, and C. K. Yuan. John Musser graciously provided access to some data from Programmableweb.com.

I would never have written this book without the inspiration I drew from my former colleagues at UC Berkeley. I want to particularly thank Chris Ashley, Isaac Mankita, and Susan Stone, who helped me persevere by listening patiently to my exuberant book talk over lunch or tea. Thom King and Tom Schirmer generously shared their knowledge of data architecture and software development. Shifra Gaman and Rich Meyer provided tons of technical and moral support. Thanks to David Greenbaum for his supportive attitude toward my teaching work and to Rick Jaffe, a colleague who took my class and who has been tirelessly championing my teaching to others. Sara Leavitt and Aron Roberts generously helped me learn about iCalendar and the Berkeley Events Calendar.

The School of Information at UC Berkeley has been an important intellectual home for the past several years. I'm grateful for the opportunity to teach my course "Mixing and Remixing Information" and the collegiality of the faculty and staff. Most of all, I want to thank my students, who gave me their attention and created wonderful projects and in turn inspired me. Thanks also to my teenage summer students for creating their mashup projects in a six-week sprint and teaching me about music mashups.

Thanks to the many people at Apress who made this book a much better product than what I alone could have written. In addition to consistently savvy editorial judgment, Matt Moodie provided me with just the right amount of encouragement in the last months of writing. Richard Dal Parto provided able project management, while Grace Wong pinch-hit in that capacity. Thanks also to production editors Laura Esterman and Lori Bring, senior production editor Kelly Winkquist, and copy editor Kim Wimpsett for their work in turning code and text into a publishable form. I'm grateful to Tina Nielsen who was one of the first people I dealt with at Apress. A special thanks goes to Chris Mills, who, as my first editor, got me off to a solid start in the writing process with his enthusiasm and detailed feedback.

I learned much from the insightful comments of John Watson, who served as technical reviewer, as well as from his very cool Flickr mashups!

I would like to thank my friends, who were excited for me and cheered me on. My family, as always, was there for me; thank you so much for loving me and believing in me as I took on this huge project.

And finally, I thank my wife, Laura, for everything she did for me while I wrote this book: her love, encouraging words, and wise counsel; her listening to my ideas-in-process; her editing; and her sacrificial willingness to free up time for me to write. It's time for us to take that long-postponed vacation!

Introduction

How many times have you seen a web site and said, “This would be exactly what I wanted—if only . . .” If only you could combine the statistics here with data from your company’s earnings projections. If only you could take the addresses for those restaurants and plot them on one map. How often have you entered the date of a concert into your calendar with a single click instead of retyping? How often do you wish that you could make all the different parts of your digital world—your e-mail, your word processor documents, your photos, your search results, your maps, your presentations—work together more seamlessly? After all, it’s all digital and malleable information—shouldn’t it all just fit together?

In fact, below the surface, all the data, web sites, and applications you use could fit together. This book teaches you how to forge those latent connections—to make the Web your own—by remixing information to create your own mashups. A *mashup*, in the words of the Wikipedia, is a web site or web application “that seamlessly combines content from more than one source into an integrated experience.”¹ Learning how to draw content from the Web together into new integrated interfaces and applications, whether for yourself or for other others, is the central concern of this book.

Let’s look at a few examples to see how people are remixing data and services to make something new and useful:

- Housingmaps.com brings together Google Maps and the housing and rental listings from Craigslist.com. Note that it was invented by neither Google nor Craigslist but by an individual programmer, Paul Radamacher. Housingmaps.com adds to the functionality of Craigslist, which will show you on a map where a specific listing is located but not all the rentals or houses in an area.²
- Google Maps in Flickr (GMiF) brings together Flickr pictures, Google Maps, Google Earth, and the Firefox browser via Greasemonkey.³
- The Library Lookup bookmark is a JavaScript bookmarklet that connects Amazon.com and your local library catalog.⁴

1. [http://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)), accessed as 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>

To create your own mashups and customize the Web, you will look at these examples in greater detail, in addition to many other examples large and small, in this book. You can solve countless specific problems by remixing information. Here are some examples of techniques you will learn in this book:

- Taking a book you found on Amazon.com and instantly locating it in your local library
- Synthesizing a single news feed from many news sources through Yahoo! Pipes
- Posting Flickr photos to blogs with a click of a button
- Displaying your photos in Google Maps and Google Earth
- Using special extensions to Firefox to learn how to program Google Maps
- Inserting extra information into web pages with Greasemonkey
- Plotting stories from your favorite news source (such as the *New York Times*) on a map
- Making your own web site remixable so that others can create mashups from your content
- Creating Google calendars from your event calendars from the Web
- Storing and retrieving your files from online storage (S3)
- Creating an online spreadsheet from your Amazon.com wishlist
- Recognizing and manipulating data embedded in web pages
- Adding an event listed on the Web to your calendar and e-mailing it to other people with one mouse click
- Building web search functionality into your own web applications
- Republishing word documents that are custom-formatted for your web site

Mashups are certainly hot right now, which is interesting because it makes you part of a shared undertaking, a movement. Mashups are fun and often educational. There's delight in seeing familiar things brought together to create something new that is greater than the sum of its parts. Some mashups don't necessarily ask to be taken that seriously. And yet mashups are also powerful—you can get a lot of functionality without a lot of effort. They might not be built to last forever, but you often can get what you need from them without having to invest more effort than you want to in the first place.

The Web 2.0 Movement

The Web 2.0 bandwagon is an important reason why mashups are popular now. Mashups have been identified explicitly (under the phrases “remixable data source” and “the right to remix”) by Tim O'Reilly in “What is Web 2.0?”⁵ Added to this, we have the development of what might be accurately thought of as “Web 2.0 technologies/mind-sets” to remix/reuse data, web services,

5. <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>

and micro-applications to create hybrid applications. Recent developments bring us closer to enabling users to recombine digital content and services:

- Increasing availability of XML data sources and data formats in business, personal, and consumer applications (including office suites)
- Wide deployment of XML web services
- Widespread current interest in data remixing or mashups
- Ajax and the availability of JavaScript-based widgets and micro-applications
- Evolution of web browsers to enable greater extensibility (for example, Firefox extensions and Greasemonkey scripts)
- Explosive growth in “user-generated content” or “lead-user innovation”
- Wider conceptualization of the Internet as a platform (“Web 2.0”)
- Increased broadband access

These developments have transformed creating mashups from being technically challenging to nearly mainstream. It is not that difficult to get going, but you need to know a bit about a fair number of things, and you need to be playful and somewhat adventurous.

Will mashups remain cutting-edge forever? Undoubtedly, no, but not because they will prove to be an irrelevant fad but because the functionality we see in mashups will eventually be subsumed into the ordinary “what-we-expect-and-think-has-always-been-there” functionality of our electronic society.

Moreover, mashups reflect deeper trends, even the deepest trends of human desire. As the quality, quantity, and diversity of information grow, users long for tools to access and manage this bewildering array of information. Many users will ultimately be satisfied by nothing less than an information environment that gives them seamless access to any digital content source, handles any content type, and applies any software service to this content. Consider, for example, what a collection of bloggers expressed as their desires for next-generation blogging tools:⁶

Bloggers want tools that are utterly simple and allow them to blog everything that they can think, in any format, from any tool, from anywhere. Text is just the beginning: Bloggers want to branch out to multiple media types including rich and intelligent use of audio, photos, and video. With input, having a dialog box is also seen as just a starting place for some bloggers: everything from a visual tool to easy capture of things a blogger sees, hears, or reads point to desirable future user interfaces for new generations of blogging tools.

Mashups are starting to forge this sought-after access and integration of data and tools—not only in the context of blogging but also to any point of interaction between users and content.

6. http://www.cadence90.com/blogs/2004_03_01_nixon_archives.html#107902918872392913

Overall Flow of the Book

A central question of this book is, how can both nontechnical end users and developers recombine data and Internet services to create something new for their own use for and for others? Although this book focuses primarily on XML, web services, and the wide variety of web applications, I'll also cover the role played by desktop applications and operating systems.

The Book's Structure

The following is a breakdown of the parts and chapters in this book:

- **Part 1**, “Remixing Information Without Programming,” introduces mashups without demanding programming skills from you and teaches skills for deconstructing applications for their remix potential.
 - Chapter 1, “Learning from Specific Mashups,” analyzes in detail a selection of mashups/remixes (specifically, Housingmaps.com, Google Maps in Flickr, and the LibraryLookup bookmarklet) to get you oriented to mashups in general and to some general themes we will continually revisit throughout the book.
 - Chapter 2, “Uncovering the Mashup Potential of Web Sites,” analyzes Flickr (as our primary extended example) for what makes it the remix platform *par excellence* for learning how to remix a specific application and exploit features that make it so remixable. We compare and contrast Flickr with other remixable platforms such as del.icio.us, Google Maps, and Amazon.com.
 - Chapter 3, “Understanding Tagging and Folksonomies,” covers tagging. Tagging, which allows users to attach words to pictures, and websites—almost anything on the Web—is the glue that holds many things together, both within and across websites. This chapter illustrates how tags are used in Flickr, del.icio.us, and Technorati and discusses how to create interesting tag-centric mashups, how people are “hacking” the tagging system to create ad hoc databases, and how tags relate to other classification systems.
 - Chapter 4, “Working with Feeds, RSS, and Atom,” presents RSS and Atom, perhaps the most widespread dialects of XML, as both a potent technology for remixing in its own right and also as a specific way to learn about XML more generally. Not to be missed are the sections on the various RSS/Atom-related formats and their significance for information remix. The chapter includes a tutorial on using Yahoo! Pipes to filter and synthesize feeds.
 - Chapter 5, “Integrating with Blogs,” uses Flickr’s integration with weblogs as a jumping-off point for an exploration of weblogs and wikis and their programmability. Integration with blogging is an important topic since blogs represent a type of remixing in a narrative, as opposed to data-oriented remixing via tags and the straight RSS so far discussed. A brief discussion of integration with wikis concludes the chapter.

- **Part 2**, “Remixing a Single Web Application Using Its API,” concentrates on teaching the broad classes of web-based APIs by studying exemplars of each class.
 - Chapter 6, “Learning Web Services APIs Through Flickr,” studies Flickr in detail. In addition to being an exemplar for a range of nonprogramming remixing techniques in Part 1, Flickr is also an excellent playground for learning XML web services. This chapter will show you how to use the Flickr API, looking first at how to make a simple call to the API, next looking at how to make sense of the entire variety of calls available, and then generalizing to handle authentication.
 - Chapter 7, “Exploring Other Web APIs,” explains commonalities and contrasts among various API providers, specifically those between Flickr and other systems, and surveys the types of services available and how to think about the sheer range of APIs. You will learn how to call REST, XML-RPC, and SOAP-based services. This chapter looks at sites, such as Programmableweb.com, that document these various APIs and the challenges faced in doing so.
 - Chapter 8, “Learning Ajax/JavaScript Widgets and Their APIs,” describes the other large class of web application remixability: those of JavaScript-based widgets, many of which are Ajax applications. This chapter contrasts old-style web applications with Ajax approaches through specific examples in Flickr and other applications and introduces the Yahoo! UI Library, a specific JavaScript widget library to demonstrate how to program widgets. You will also learn how to use the Firebug Firefox extension and the JavaScript Shell to learn about JavaScript. The chapter concludes with an introduction to using Greasemonkey.
- **Part 3**, “Making Mashups,” is the heart of the book; it’s a discussion of how to use what you learned in Parts 1 and 2 to create mashups.
 - Chapter 9, “Moving from APIs and Remixable Elements to Mashups,” analyzes mashups and their relationship to APIs through studying a series of specific problems for which mashups can provide useful solutions. The chapter looks at how you can track books, real estate, airfare, and current events by combining various APIs. You will learn how to use Programmableweb.com to analyze these problems.
 - Chapter 10, “Creating Mashups of Several Services,” teaches you how to write mashups by providing a detailed example that you’ll build from the ground up: a mashup of geo-tagged Flickr photos and Google Maps using first the Google Maps API and then the Google Mapplets API.
 - Chapter 11, “Using Tools to Create Mashups,” discusses tools that have been developed to make creating mashups easier than by using traditional web programming techniques. This chapter walks you through using one of these tools—the Google Mashup Editor—and briefly surveys other tools.
 - Chapter 12, “Making Your Web Site Mashable,” shifts the focus of the book briefly from the consumption to the production of data and APIs. This chapter is a guide to content producers who want to make their web sites friendly to mashups. That is, this chapter answers the question, how would you as a content producer make your digital content most effectively remixable and mashable to users and developers?

- **Part 4**, “Exploring Other Mashup Topics,” covers how to remix and integrate specific classes of applications, using the core conceptual framework of Parts 1 to 3 to guide the discussion.
 - Chapter 13, “Remixing Online Maps and 3D Digital Globes,” covers popular online maps and virtual globes, offering examples of map-based mashups. You’ll learn about making maps without programming and data exchange formats (GeoRSS and KML), and then you’ll turn to the various APIs: Google Maps, Yahoo! Maps, and Microsoft Maps. I’ll also cover geocoding American and non-American addresses. The chapter closes with a discussion of Google Earth, its relationship to KML, and how to display Flickr photos via KML.
 - Chapter 14, “Exploring Social Bookmarking and Bibliographic Systems,” covers how social bookmarking responds to a fundamental challenge—the job of keeping found things found on the Web, which, at a basic level, is done through URLs, but you’ll learn about other digital content such as images and data sets. Social bookmarking is interesting not only for the extensibility/remixability being built into these systems but also for the insight it offers into other systems. This chapter walks you through a select set of social bookmarking systems and their APIs, as well as discusses interoperability challenges among these systems. The chapter shows how to create a mashup of Flickr and del.icio.us.
 - Chapter 15, “Accessing Online Calendars and Event Aggregators,” shows what data you can get in and out of calendars without programming (using iCalendar and XML feeds), how to program individual calendars (using Google Calendar and 30boxes), and how to program individual event aggregator APIs (using Upcoming.yahoo.com and Eventful.com). The chapter concludes with a mashup of a public events calendar with Google Calendar.
 - Chapter 16, “Using Online Storage Services,” surveys the potentially important and growing area of online storage solutions and shows the basics of using Amazon S3.
 - Chapter 17, “Mashing Up Desktop and Web-Based Office Suites,” shows how to do some simple parsing in ODF and OpenXML, demonstrates how to create a simple document in both ODF and OpenXML, explains some simple scripting of Microsoft Office and OO.o, and concludes with a mashup of Google Spreadsheets and Amazon.com web services.
 - Chapter 18, “Using Microformats and RDFa As Embeddable Data Formats,” studies two answers to the problem of how to embed information in web pages that is easy to understand by both humans and computer programs: microformats and RDFa. You will learn how to use and program the Operator Firefox extension to recognize and manipulate microformats.
 - Chapter 19, “Integrating Search,” shows how to use the Google Ajax Search API, Yahoo! Search APIs, and Microsoft Live.com search; the chapter also introduces OpenSearch and the Google Desktop HTTP/XML gateway.

Intended Audience

This book is accessible to a wide range of readers, including those who are curious about Web 2.0 applications and those who want to know more about the technical underpinnings of it. The technical prerequisites are a good understanding of HTML, basic CSS, and basic JavaScript. References to appropriate background materials will be provided. In this book, most of the server-side code is presented in PHP. Some code is in Python.

At the same time, experienced developers will also be able to learn much from the book. Although there will be a breadth of coverage, I will strive to state deep, essential facts about the technologies in question (with respect to their applicability to remix)—aspects that might not be obvious at first glance.

Information remixing can easily come across as a confusing grab bag of techniques. Beginners have a hard time understanding the significance of XML, web services, Ajax, COM, and metadata for remixing data. It is not that difficult to get going, but you need to know a bit about a fair number of different topics, and you need to be playful and somewhat adventurous. Usually these topics are found scattered throughout a large selection of books; this book is the guide to show you where to begin.

Updates

Please go to <http://mashupguide.net> to find updates and supplementary materials for this book.

