
Table of Contents

Foreword	xi
Preface	xiii
1. The Programmable Web and Its Inhabitants	1
Kinds of Things on the Programmable Web	4
HTTP: Documents in Envelopes	5
Method Information	8
Scoping Information	11
The Competing Architectures	13
Technologies on the Programmable Web	18
Leftover Terminology	20
2. Writing Web Service Clients	23
Web Services Are Web Sites	23
del.icio.us: The Sample Application	26
Making the Request: HTTP Libraries	29
Processing the Response: XML Parsers	38
JSON Parsers: Handling Serialized Data	44
Clients Made Easy with WADL	47
3. What Makes RESTful Services Different?	49
Introducing the Simple Storage Service	49
Object-Oriented Design of S3	50
Resources	52
HTTP Response Codes	54
An S3 Client	55
Request Signing and Access Control	64
Using the S3 Client Library	70
Clients Made Transparent with ActiveResource	71
Parting Words	77

4.	The Resource-Oriented Architecture	79
	Resource-Oriented What Now?	79
	What's a Resource?	81
	URIs	81
	Addressability	84
	Statelessness	86
	Representations	91
	Links and Connectedness	94
	The Uniform Interface	97
	That's It!	105
5.	Designing Read-Only Resource-Oriented Services	107
	Resource Design	108
	Turning Requirements Into Read-Only Resources	109
	Figure Out the Data Set	110
	Split the Data Set into Resources	112
	Name the Resources	117
	Design Your Representations	123
	Link the Resources to Each Other	135
	The HTTP Response	137
	Conclusion	140
6.	Designing Read/Write Resource-Oriented Services	143
	User Accounts as Resources	144
	Custom Places	157
	A Look Back at the Map Service	165
7.	A Service Implementation	167
	A Social Bookmarking Web Service	167
	Figuring Out the Data Set	168
	Resource Design	171
	Design the Representation(s) Accepted from the Client	183
	Design the Representation(s) Served to the Client	184
	Connect Resources to Each Other	185
	What's Supposed to Happen?	186
	What Might Go Wrong?	187
	Controller Code	188
	Model Code	205
	What Does the Client Need to Know?	209
8.	REST and ROA Best Practices	215
	Resource-Oriented Basics	215

The Generic ROA Procedure	216
Addressability	216
State and Statelessness	217
Connectedness	218
The Uniform Interface	218
This Stuff Matters	221
Resource Design	227
URI Design	233
Outgoing Representations	234
Incoming Representations	234
Service Versioning	235
Permanent URIs Versus Readable URIs	236
Standard Features of HTTP	237
Faking PUT and DELETE	251
The Trouble with Cookies	252
Why Should a User Trust the HTTP Client?	253
9. The Building Blocks of Services	259
Representation Formats	259
Prepackaged Control Flows	272
Hypermedia Technologies	284
10. The Resource-Oriented Architecture Versus Big Web Services	299
What Problems Are Big Web Services Trying to Solve?	300
SOAP	300
WSDL	304
UDDI	309
Security	310
Reliable Messaging	311
Transactions	312
BPEL, ESB, and SOA	313
Conclusion	314
11. Ajax Applications as REST Clients	315
From AJAX to Ajax	315
The Ajax Architecture	316
A del.icio.us Example	317
The Advantages of Ajax	320
The Disadvantages of Ajax	320
REST Goes Better	322
Making the Request	323
Handling the Response	324
JSON	325

Don't Bogart the Benefits of REST	326
Cross-Browser Issues and Ajax Libraries	327
Subverting the Browser Security Model	331
12. Frameworks for RESTful Services	339
Ruby on Rails	339
Restlet	343
Django	354
A. Some Resources for REST and Some RESTful Resources	365
Standards and Guides	365
Services You Can Use	367
B. The HTTP Response Code Top 42	371
Three to Seven Status Codes: The Bare Minimum	372
1xx: Meta	373
2xx: Success	374
3xx: Redirection	377
4xx: Client-Side Error	380
5xx: Server-Side Error	387
C. The HTTP Header Top Infinity	389
Standard Headers	390
Nonstandard Headers	404
Index	409