

Symbols and Numbers

- , (commas)
 - scoping information, 118–121
 - URI design and, 233
- / (forward slash)
 - Django “house style” and, 356
 - using XPath, using, 24
- // XPath expressions, 9
- ; (semicolons) scoping information, 118–121
- [] (square brackets), using XPath, 24
- 100 “Continue” response code, 373
- 101 “Switching Protocols” response code, 373
- 200 “OK” response code, 54, 137, 140, 374
 - database-backed control flow, 274
 - modifying resources, 186
- 201 “Created” response code, 374
 - creating resources, 186
 - database-backed control flow, 274
- 202 “Accepted” response code, 229, 375
- 203 “Non-Authoritative Information” response code, 375
- 204 “No Content” response code, 375
- 206 “Partial Content” response code, 376
- 207 “Multi-Status” response code, 376
- 300 “Multiple Choices” response code, 377
- 301 “Moved Permanently” response code, 274, 372, 377
- 302 “Found” response code, 378
- 303 “See Other” response code, 139, 378
- 304 “Not Modified” response code, 245, 379
- 305 “Use Proxy” response code, 379
- 306 “Unused” response code, 380
- 307 “Temporary Redirect” response code, 380
- 400 “Bad Request” response code, 140, 381
 - database-backed control flow, 273
 - restrictions and, 187
- 401 “Unauthorized” response code, 156, 187, 239, 381
- 402 “Payment Required” response code, 381
- 403 “Forbidden” response code, 382
- 404 “Not Found” response code, 54, 139, 187, 373, 382
 - database-backed control flow, 274
- 405 “Method Not Allowed” response code, 283, 382
- 406 “Not Acceptable” response code, 383
- 407 “Proxy Authentication Required” response code, 383
- 408 “Request Timeout” response code, 383
- 409 “Conflict” response code, 156, 373, 383
 - database-backed control flow, 274
 - unauthorized access and, 187
- 410 “Gone” response code, 274, 373, 384
- 411 “Length Required” response code, 384
- 412 “Precondition Failed” response code, 385
- 413 “Request Entity Too Large” response code, 385
- 414 “Request-URI Too Long” response code, 386
- 415 “Unsupported Media Type” response code, 156, 188, 386
 - database-backed control flow, 273
- 416 “Requested Range Not Satisfiable” response code, 386
- 417 “Expectation Failed” response code, 386
- 500 “Internal Server Error” response code, 54, 140, 157, 372, 387
- 501 “Not Implemented” response code, 387
- 502 “Bad Gateway” response code, 387

We’d like to hear your suggestions for improving our indexes. Send email to index@oreilly.com.

503 “Service Unavailable” response code, 140, 157, 388
504 “Gateway Timeout” response code, 388
505 “HTTP Version Not Supported” response code, 388
“Abusing Amazon Images” (Gertler, Nat), 107
“Universal Resource Identifiers-Axioms of Web Architecture” (Berners-Lee, Tim), 82

A

Accept request header, 390
Accept-Charset request header, 390
Accept-Encoding request header, 30, 390
 compression algorithms and, 243
Accept-Language request header, 93, 391
Accept-Ranges request header, 391
“Accepted” 202 response code, 229, 375
access control, 64–70
“Access Key ID” (Amazon), 56
access policies, 69
ActionScript, 38, 316
 XML parsers and, 44
ActiveRecord, 183
ActiveResource, 25
 clients, making transparent with, 71–77
acts_as_taggable plugin, 168, 205
ad hoc XHTML, 268
addressability of URIs, 84–86, 216, 221
Age request header, 391
Ajax, 86
 advantages/disadvantages, 320
 architecture, 316
 cross-browser issues and, 327
 request proxying, 332
 requests, 323
 responses, handling, 324
 REST clients, as, 315
AllegroServe web server library (Lisp), 38
Allow request header, 392
Amazon Web Services, 3, 50
 addressability, 85
 S3, 13, 52
 client library, using, 70
 wrappers, 25
Apache Tomcat, 344
API keys, 144
APP (Atom Publishing Protocol), 13, 49, 275–281
 collections, describing, 295

application forms, 284
application state, 90, 218
application/atom+xml media types, 263
application/json media type, 266
application/x-www-form-urlencoded media type, 266
application/xhtml+xml media type, 259
application/xml media type, 268, 269
applications and web interfaces, 254–257
applicaton/xhtml+xml
 ad hoc XHTML, 268
Architecture of the World Wide Web, 81
Asynchronous JavaScript And XML (see Ajax)
asynchronous operations, 228
AsyncWeb, 344
Atom, 168, 185, 263–265
“Atom Authentication” (Pilgrim, Mark), 241
atom-tools Ruby gem, 168
“authenticated” messages, 154
authenticated-read access policy, 70
authentication (HTTP), 146
 authorization and, 238–243
authorization, 146–147
 authentication and, 238
 unauthorized access and, 187
Authorization request header, 30, 239, 392
 S3 and, 64
 unauthorized access and, 187
 WSSE HTTP authentication, 241
authorization token, 254
AWS::S3, 50, 55
Axioms of Web Architecture (Berners-Lee, Tim), 236

B

“Bad Gateway” 502 response code, 387
“Bad Request” 400 response code, 140, 381
 database-backed control flow, 273
 restrictions and, 187
base 64 encoding, 238
Basic HTTP authentication, 30, 146, 238
batch operations, 230
Beautiful Soup XML parser, 42
Berners-Lee, Tim, 82, 236
best practices for REST, 215–258
Big Web Services, 5, 299–314
binary documents, 279
bookmarks, 26
 controllers for, 175

- management for, 180
- BPEL (Business Process Execution Language), 313
- browser issues, 327
- buckets, 4
 - access policies, 69
- buckets (S3), 50
- Builder::XmlMarkup (Ruby), 203
- Bunardzic, Alex, 81
- bundlers controller, 179
- Business Process Execution Language (BPEL), 313

C

- C programming language, 38
 - XML parsers and, 44
- C#, 36
 - XML parsers and, 43
- C++ programming language, 38
 - XML parsers and, 44
- Cache-Control header, 248, 392
- caching, 247–249
- calendar controller, 178
- __call__ method, 360
- category documents (APP), 278
- character encoding, 270
- class HTML attribute, 260
- client-side errors, 380–386
- clients, 23–48
 - ActiveResource, making transparent with, 71–77
 - Python, using for, 76
 - representations, 150–152, 159–161
 - S3, 55–64
 - library, using, 70
 - writing, 30, 346–349
- CLR (Common Language Runtime), 36, 43
- collections (APP), 275
- commas (,)
 - scoping information, 118–121
 - URI design and, 233
- Common Language Runtime (CLR), 36, 43
- Common Lisp, 38
 - XML parsers, 44
- compression, 243
- conditional GET, 138, 189, 244–247
 - caching and, 247
- “Conflict” 409 response code, 156, 373, 383
 - database-backed control flow, 274

- unauthorized access and, 187
- connectedness, 94–96, 218, 223
 - service versioning, 235
- Connection request header, 392
- content negotiation, 92
- Content-Encoding header, 243, 393
- Content-Language request header, 393
- Content-Length request header, 393
- Content-Location header, 217, 249, 394
- Content-MD5 request header, 394
- Content-Range request header, 394
- Content-Type header, 7, 394
 - compression algorithms and, 243
 - encodings and, 271
 - HTTP response and, 137
 - outgoing representations and, 234
- S3, 61
- “Continue” 100 response code, 373
- control flows (prepackaged), 272–284
- controllers, 173, 175–179, 340
 - code, 188–205
- Cookie request header, 405
- cookies, 89, 252
- CPAN
 - XML parsers and, 44
- “Created” 201 response code, 374
 - creating resources, 186
 - database-backed control flow, 274
- cross-browser issues, 327
- Crypt:SSLeay module, 38
- CSS, 39
- curl, 37

D

- data sets, 110–112, 148, 157, 168–171
 - resources, splitting into, 157
- databases, 169, 186
 - ActiveRecord and, 183
 - control flow and, 273–275
- Date request header, 395
- DCMI (Dublin Core Metadata Initiative), 267
- dd tag (HTML), 131
- Debian GNU/Linux, installing net/https library, 31
- del.icio.us web service, 26–28
- DELETE method, 8, 29, 97
 - ActiveResource clients and, 75
 - APP resources and, 281
 - caching and, 249

- faking, 251
- S3, 52
- S3::Bucket#delete method, 59
- safety and idempotence, 102
- uniform interface and, 219
- user controllers and, 174
- UsersController and, 196
- web applications and web services and, 342

DELETE statement (SQL), 75

Digest HTTP authentication, 30, 146, 239–241

Django (Python), 167, 339, 355–364

- implementing resources as views, 357–364

dl HTML tag, 260

document (entity-body), 6

document-based protocol, 5

Dojo, 45, 330

DOM parsers, 39

DOMIT! DOM parser (PHP), 43

Dublin Core Metadata Initiative (DCMI), 267

E

ECMAScript standards, 45

ElementTree (Python), 42

encoding issues, 270

Enterprise Service Bus (ESB), 313

entity-body, 6, 150

- batch operations and, 230
- form-encoding and, 151
- HTTP libraries and, 29
- Look-Before-You-Leap (LBYL) requests and, 249
- PUT/DELETE, faking, 252
- scoping informations and, 12
- XML documents and, 28, 38

entries (Atom lists), 263

EntriesController class, 340

eRDF, 267

error (status codes)

- client-side, 380–386
- server-side, 387

ESB (Enterprise Service Bus), 313

ETag HTTP header, 30, 187, 395

- conditional GET and, 189, 245

Expat XML parser, 44

Expect request header, 395

“Expectation Failed” 417 response code, 386

expiration dates, signing URIs, 69

Expires request header, 396

Extensible Open XHTML Outlines (XOXO), 262

F

federation, 311

feeds (Atom lists), 263

Firefox, 147

Flash, 38

Flickr

- API, 8, 16
- statelessness and, 90
- user accounts and, 144

FOAF, 268

“Forbidden” 403 response code, 382

form-encoding, 183

- key-value pairs, 38, 266

forms (HTML)

- encoded representation of user accounts, 183
- form-encoded key-value pairs and, 266
- hypermedia and, 284
- linking resources and, 135
- user accounts and, 145
- XHTML 4, 287

forward slash (/)

- Django “house style” and, 356
- XPath, using, 24

“Found” 302 response code, 378

4Suite, 42

Framework-specific serialization formats, 268

frameworks for RESTful services, 339–364

From request header, 396

G

“Gateway Timeout” 504 response code, 388

GData, 13, 281

gem program, 31

geo microformat specification, 263

Gertler, Nat, 107

GET method, 6, 29, 97

- ActiveResource clients and, 75
- APP resources and, 281
- conditional HTTP, 138, 189
- database-backed control flow and, 274
- open-uri library and, 31
- partial, 250
- read-only resources and, 109

- Ruby controllers and, 341
- S3, 52, 59
- safety and idempotence, 102
- scoping information, 12
- uniform interface and, 218
- user controllers and, 174

Gmail, 86

- Ajax and, 315, 326

“Gone” 410 response code, 274, 373, 384

Google, 10

- Calendar, 254
- GData, 281
- links and connectedness, 94
- maps, 127
- representations and, 93
- Resource-Oriented Architecture and, 86
- SOAP and, 301
- Web Accelerator and, 103

Gourley, David, 247

Gregorio, Joe, 33, 109

gs:doGoogleSearch remote procedure, 301

GUI elements, 317

gzip, 243

H

Hadley, Marc, 293

hAtom microformat specification, 263

hCalendar microformat specification, 261, 268

hCard microformat specification, 261, 268

HEAD method, 29, 98

- caching and, 247
- read-only resources and, 109
- S3, 52, 62
- safety and idempotence, 102
- uniform interface and, 219

headers, 389–407

- nonstandard, 404–407
- standard, 390–404

hierarchy into path variables, 118–121

High REST, 21

Host request header, 396

How to create a REST Protocol (Gregorio, Joe), 109

hpricot gem, 40

href attribute (link tag), 286

hResume microformat specification, 263

hReview microformat specification, 263

HTML, 4, 289

- linking resources, 135
- XHTML and, 260

HTTP, 4, 5–7, 18

- authentication, 146, 238–243
 - Basic authentication, 146, 238
- data sets and, 112
- Digest authentication, 146, 239–241
- encodings, 271
- libraries, 29
- methods, 97
- RPC-style architectures and, 14
- sessions, 89
- standard features of, 237–251
 - WSSE authentication, 30, 146, 241

HTTP Basic authentication, 146

HTTP response codes (see response codes (HTTP))

“HTTP Version Not Supported” 505 response code, 388

HTTP+POX (HTTP plus Plain Old XML), 21

HTTP: The Definitive Guide (Totty, Gourley), 247

HttpClient (Java), 34

httplib2 (Python), 33

HTTPS, 146

- certificates, 29

HttpURLConnection class, 344

HTTPWebRequest, 36

http_authentication Ruby plugin, 168

Hybrid architectures (RPC), 16

hypermedia, 95, 154, 161

- descriptions, 212
- natural-language description and, 210
- technologies, 284–298
 - WADL and, 290–298

hypermedia as engine of application state (see connectedness)

I

iCalendar, 268

idempotence, 102–104

- uniform interface, 219

If-Match HTTP header, 396

If-Modified-Since HTTP header, 30, 138, 245, 397

If-None-Match HTTP header, 138, 397

- conditional GET and, 245

If-Range HTTP header, 397

If-Unmodified-Since header, 398

- image/svg+xml media type, 265
- incoming representations, 234
- INSERT statement (SQL), 75
- “Internal Server Error” 500 response code, 54, 140, 157, 372, 387
- ISO 8859-1 encoding, 270
- itemsPerPage element (OpenSearch), 265

J

- JAR files, 354
- Java, 34, 316
 - XML parsers and, 43
- java.net.HttpURLConnection HTTP client, 34
- java.net.URL object, 34
- JavaScript, 37
 - Ajax and, 316
 - on Demand, 333
 - XML parsers, 43–47
- JavaScript Object Notation (see JSON)
- javax.xml.* package, 43
- javax.xml.stream package, 43
- jbucket, 25
- Jetty, 344
- JSON (JavaScript Object Notation), 4, 44–47, 125, 266
 - Ajax and, 325
 - bookmarks, representing, 184
 - encoding and, 272
- json Ruby gem, 46

K

- key-value pairs, 183, 266
- keys (S3), 51

L

- language (natural) description, 210
- Last-Modified HTTP header, 30, 138, 187, 398
 - conditional GET and, 189, 245
- LBYL (Look-Before-You-Leap) requests, 249
- “Length Required” 411 response code, 384
- li HTML tag, 260
- libcurl, 37
- libgmail library, 315
- libopenssl-ruby, 31
- libraries
 - HTTP, 29
 - S3, using, 70

- standard, 1
- libwww-perl (LWP), 38
- libxml2 library, 40
 - PHP XML parsers and, 43
- limit parameter, 190
- link tag, 286
- links, 58, 94–96
 - hypermedia and, 284
 - resources to existing resources, 154–155, 161–164
 - S3 clients and, 224
 - XHTML 4, 285
- Linux, installing net/https library, 31
- Lisp, 38
 - XML parsers, 44
- Location response header, 249, 377, 399
- Look-Before-You-Leap (LBYL) requests, 249
- Lovett, Chris, 43
- Low REST, 21
- LWP (libwww-perl), 38

M

- Max-Forwards header, 399
- media types, 7
- members (APP), 275
- metadata, 92
- method information, 8
- “Method Not Allowed” 405 response code, 382
- methods, 8–11
 - S3, 53
- microformats with XHTML, 261–263
- model classes, 205–209
- “Moved Permanently” 301 response code, 274, 372, 377
- “Multi-Status” 207 response code, 376
- “Multiple Choices” 300 response code, 377
- MySQL, 169

N

- names for resources, 117–123
- natural-language description, 210
- .NET Common Language Runtime (CLR), 36, 43
- net/http library, 31
- Net::HTTP class, 32, 39
- NetworkCredential, 36
- Nielsen, Jakob, 236

- “No Content” 204 response code, 375
- Noelios Restlet Engine (NRE), 344
- “Non-Authoritative Information” 203 response code, 375
- “Not Acceptable” 406 response code, 383
- “Not Found” 404 response code, 54, 139, 187, 373, 382
 - database-backed control flow, 274
- “Not Implemented” 501 response code, 387
- “Not Modified” 304 response code, 245, 379
- Nottingham, Mark, 283
- NRE (Noelios Restlet Engine), 344

O

- object (RDF assertion), 267
- object-oriented design of S3, 50–51
- object-relational mapping (ORM), 355
- objects, 4
 - S3, 50, 61–64
- “OK” 200 response code, 54, 137, 140, 274, 374
 - modifying resources, 186
- open-uri library, 30, 39
- OpenSearch, 265
- “Method Not Allowed” 405 response code, 283
- OPTIONS method, 29, 98
 - uniform interface and, 219
- org.w3c.dom.* package, 43
- org.xml.sax.* package, 43
- ORM (object-relational mapping), 355
- outgoing representations, 234
- overloaded POST, 101, 220
 - PUT/DELETE, faking, 252
 - safety and idempotence, 219
 - URI design and, 233

P

- params parameter, using list item resources and, 189
- Park Place, 77
- parsers (XML), 38
- “Partial Content” 206 response code, 376
- path variables
 - designing URIs, 233
 - encoding hierarchy into, 118–121
- paths (URI), 6
- “Payment Required” 402 response code, 381

- Perl, 38
 - XML parsers, 44
- permanent URIs, 236
- PHP, 37
 - XML parsers, 43
- Pilgrim, Mark, 241
- places as resources, 114
- PNG format, 126
- POE (POST Once Exactly), 283
- POE request header, 405
- POE-Links header, 405
- POST method, 8, 29, 99–102, 159
 - ActiveResource clients and, 75
 - APP resources and, 281
 - caching and, 249
 - objects, creating, 176
 - Once Exactly (POE), 283–284
 - overloading, 101, 219, 220, 233
 - resources, creating/appending, 274
 - Ruby controllers and, 341
 - S3, 52
 - subordinate resources, creating, 160
 - uniform interface and, 219
 - user controllers and, 174
 - versus PUT, 220
 - web applications and web services and, 343
- postNewAtomMember method (Atom), 296
- Pragma header, 399
- “Precondition Failed” 412 response code, 385
- prepackaged control flows, 272–284
- privacy (user accounts), 146–147
- private access policy, 70
- private keys (S3 requests), 65
- programmable web, 1–21
 - technologies on, 18–20
- Prototype, 329
- “Proxy Authentication Required” 407 response code, 383
- proxy caches, 85
- Proxy-Authenticate header, 400
- Proxy-Authorization header, 400
- ProxyPass, 332
- public-read access policy, 70
- public-write access policy, 70
- pull parsers, 39
- PULL method, 8, 97
 - ActiveResource clients and, 75
 - APP resources and, 281

- caching and, 249
- database-backed control flow and, 274
- faking, 251
- HTTP libraries and, 29
- overloading
 - PUT/DELETE, faking, 252
- Ruby controllers and, 341
- S3, 52, 63
- S3::Bucket#put, 59
- safety and idempotence, 102
- uniform interface and, 219
- user controllers and, 174
- UserController and, 196
- versus POST, 99, 220
- web applications and web services and, 342

- Python
 - clients, 76
 - httplib2, 33
 - XML parsers, 42

Q

- query strings, 183
- query variables, 121–123
 - URI design and, 233

R

- Rails, 339–364
- Range request header, 400
- RDF (Resource Description Framework), 266
- read-only resource-oriented services, 107–141
- read-only web services, 17
- read/write resource-oriented services, 143–166
- readable URIs, 236
- recent bookmark controller, 178
- redirect loops, 30
- redirection, 377–380
- Referer request header, 401
- regular expressions (Django), 358
- rel attribute (link tag), 286
- rel HTML attributes, 260
- rel-license microformat specification, 261
- rel-nofollow microformat specification, 262
- rel-tag microformat specification, 262
- reliable messaging, 311
- Remote Procedure Calls (RPC), 19
- repetition model (XHTML), 136

- representations, 91–94, 216
 - addressability and, 217
 - designing, 123–134, 152, 183–185
 - entity-body, 6
 - formats, 259–272
 - incoming, 342
 - outgoing, 341
 - outgoing/incoming, 234
 - S3, 57
- “Request Entity Too Large” 413 response code, 385
- request headers, 6
- request signing, 64–70
- “Request Timeout” 408 response code, 383
- “Request-URI Too Long” 414 response code, 386
- “Requested Range Not Satisfiable” 416 response code, 386
- resource
 - design, 108
 - forms, 284
 - state, 90, 124, 217
- Resource Description Framework (RDF), 266
- Resource-Oriented Architecture (ROA), 13, 79–105, 215–258
 - addressability, 84–86
 - basics, 215
 - procedure, 216
 - read/write, designing, 143–166
 - representations and, 91
 - URIs, 81–84
 - versus Big Web Services, 299–314
- resources, 52, 143–166, 215, 340
 - connecting, 185
 - data sets, splitting into, 157
 - design, 171–183, 227–233
 - Django, implementing views, 357–364
 - relationships between, 228
 - URIs, naming resources with, 158
 - user accounts, 144–157
- resources, defining, 356
- response codes (HTTP), 7, 54, 137–141, 324, 371
 - database-backed control flow, 274
- responseXML parser (JavaScript), 43
- REST (Representational State Transfer), sending representations and, 218
- rest-open-uri library, 31
- rest-open-uri, installing, 31

Restlet (Java), 167, 339, 343–354
 services, writing, 349–354
 Retry-After response header, 401
 rev attribute (link tag), 286
 rev HTML attributes, 260
 REXML, 40–42
 REXML::document parser, 31
 ROA (see Resource-Oriented Architecture)
 routing (Rails), 340
 RPC (Remote Procedure Calls), 19
 RPC-style architectures, 14
 Ruby
 Amazon Web Services and, 3
 HTTP client libraries, 31
 on Rails, 339–343
 XML parsers and, 40–42

S

S3 (Simple Storage Service), 4, 49–77
 addressability, 85
 client, 55–64
 library, using, 70
 connectedness and, 223
 request signing and access control, 64–70
 S3::Bucket#delete method, 58
 S3::Bucket#put method, 58
 s3sh (command shell for Ruby), 4, 25
 safety (HTTP methods), 102–104
 SAX parsers, 39
 scoping information, 11
 encoding hierarchy into path variables and,
 118–121
 script tag, 335
 search results, representing lists of, 133–134
 “Secret Access Key” (Amazon), 56
 Secure Socket Layer (see SSL)
 security, 310
 “See Other” 303 response code, 139, 378
 SELECT statement (SQL), 75
 semicolons (;) scoping information, 118–121
 server-side errors, 387
 service documents (APP), 276
 “Service Unavailable” 503 response code, 140,
 157, 388
 service versioning, 235
 Service-Oriented Architecture (see SOA)
 Service-Trampled REST (STREST), 21
 services, 259–298
 Restlet, writing, 349–354

session affinity, 91
 session replication, 91
 sessions (HTTP), 89
 Set-Cookie response header, 406
 signatures, 310
 Simple Storage Service (S3), 4, 49–77
 addressability, 85
 client, 55–64
 library, using, 70
 connectedness and, 223
 request signing and access control, 64–70
 simple-http (Common Lisp), 38
 Slug request header, 406
 Snell, James, 81
 SOA (Service-Oriented Architecture), 20, 314
 SOAP, 10, 19, 300–303
 POST method, overloading, 220
 REST as a competitor to, 20
 S3 and, 53
 security and, 310
 SQL databases, 75, 273
 square brackets ([]), using XPath, 24
 SSL (Secure Sockets Layer), 311
 certificates, 29
 standard libraries, 1
 standardization, 212
 startindex element (OpenSearch), 265
 state (see statelessness)
 statelessness, 217, 222
 application state versus resource state, 90
 ROA, 86–91
 STREST (Service-Trampled REST), 21
 subject (RDF assertion), 267
 subordinate resources, 158
 creating, 99
 HTTP POST requests, creating with, 160
 Sun Web Services Developer Pack, 43
 SVG format, 126, 265
 “Switching Protocols” 101 response code, 373

T

table HTML tag, 260
 tags, 26
 management for, 181
 TagSoup XML parser, 43
 TCP/IP sockets, 28
 TE request header, 401
 “Temporary Redirect” 307 response code, 380
 terminology of web services, 4

- text/html media, 259
- text/xml media, 272
- TLS (Transport Layer Security), 311
- totalResults element (OpenSearch), 265
- Totty, Brian, 247
- to_xml (Rails), 184
 - connecting resources and, 185
 - Framework-specific serialization formats, 268
- TRACE method, 29
- Trailer response header, 402
- transactions, 231, 312
- Transfer-Encoding header, 402
- Transport Layer Security (TLS), 311
- tree-style parsers, 39
- triple (RDF assertion), 267
- trust (user accounts), 146–147

U

- UDDI, 309
- ul HTML tag, 260
- “Unauthorized” 401 response code, 239
- “Unauthorized” 401 response code, 156, 187, 381
- unauthorized access, 187
- Unicode, 270
 - parsing XML and, 42
- Uniform class (Restlet), 345
- uniform interface, 79, 104, 218–221, 222
 - exposing subsets of, 149, 158
- Universal Encoding Detector, 271
- Universal Product Codes (UPCs), 14
- Universal Resource Identifier (see URIs)
- “Unsupported Media Type” 415 response code, 156, 188, 386
 - database-backed control flow, 273
- “Unused” 306 response code, 380
- UPCs (Universal Product Codes), 14
- UPDATE statement (SQL), 75
- Upgrade request header, 402
- URI Templating, 155
- URIs (Universal Resource Identifier), 1, 15, 18, 123
 - addressability and, 216
 - bookmarks and, 26–28, 176
 - controller, 178
 - design, 233
 - Django and, 356
 - permanent versus readable, 236

- resources, naming, 149, 158
- ROAs and, 81–84
- signing, 68
- templates, 285
- using maximum length workarounds, 220
- web clients, writing, 24
- “URL as UI” (Nielsen, Jakob), 236
- urllib2 HTTP client, 33
- URLs (see URIs)
- US-ASCII encoding, 270
- “Use Proxy” 305 response code, 379
- user accounts, 144–157, 180
 - linking resources, 154–155
- user controllers, 174
- user tags controller, 177
- User-Agent request header, 403
- UsersController, 196–199
- UsersController#create Rails method, 174
- UsersController#destroy Rails method, 174
- UsersController#index Rails method, 174
- UsersController#show Rails method, 174
- UsersController#update Rails method, 174
- UTF-8 encoding, 270

V

- values (S3), 51
- Vary response header, 403
- VBScript, 316
- vCard, 268
- Via header, 404
- views, 340
 - Django, implementing resources as, 357–364
- VirtualHost class (Restlet), 346
- VoteLinks, 262

W

- W-* technologies, 19
- W3C’s HTML validator, 176
- WADL (Web Application Description Language), 20, 25, 47, 71, 308
 - hypermedia technologies and, 285, 290–298
- Warning response header, 404
- Web Accelerator, 103
- Web Application Description Language (see WADL)

- Web Hypertext Application Technology Working Group (WHATWG), 290
- web interfaces and applications, 254–257
- web service clients (see clients)
- Web Service Description Language (see WSDL)
- Web Services Developer Pack (Sun), 43
- WebDAV, 104
- WeblogsController class, 340
- WHATWG (Web Hypertext Application Technology Working Group), 290
- Windows-1252 encoding, 270
- wrapper libraries, 23
- wrappers, 25
- WS-Addressing standard, 5
- WS-ReliableMessaging, 311
- WS-Security Extension (see WSSE HTTP authentication)
- WSDL (Web Service Description Language), 10, 20, 304
 - POST method, overloading, 220
- WSSE HTTP authentication, 30, 146, 241–243
- WWW-Authenticate header, 156, 404
 - Basic authentication and, 238
 - Digest authentication and, 239
 - WSSE HTTP authentication and, 241

X

- x-amz-acl header, 69
- X-HTTP-Method-Override request header, 406
- X-WSSE request header, 406
- Xerces, 43
- XFN (XHTML Friends Network), 262
- xFolk microformat specification, 263
- XHTML, 125, 259–261, 289
 - ad hoc, 268
 - microformats, 261–263
 - XHTML 4, 285–289
- XMDP (XHTML Meta Data Profiles), 262
- XML, 4
 - ActiveRecord and, 184
 - ad hoc vocabularies, 269
 - encodings and, 271
 - parsers, 38
 - representation of user accounts, 183
 - SOAP, using, 300–303
 - XHTML and, 260
 - XML-RPC requests and, 15

- XML for <SCRIPT>, 44
- XML-RPC, 14, 19
 - POST method, overloading, 220
- xml.sax module (Python), 42
- XML::LibXML::Reader module (Perl), 44
- XML::SAX::PurePerl module (Perl), 44
- XML::Simple module (Perl), 44
- XML::XPath modules (Perl), 44
- XMLHttpRequest (JavaScript), 37, 43, 315, 323
 - cross-browser issues and, 327
- XMLPull, 43
- xml_parser_create function (PHP), 43
- XOXO (Extensible Open XHTML Outlines), 262
- XPath, 9, 28, 39, 348
 - exposition, 24
 - S3 and, 57

Y

- YAGNI (You Aren't Gonna Need It), 314
- Yahoo! web services, 13
 - searching the Web with, 23
 - user accounts and, 144