

Index

Symbols & Numerics

@ replies, 8, 9
@AusTraffic, 8
@big_ben_clock, 8
64-bit integers, using, 362
140tees Web site, 340

• A •

acceptance testing, automating, 315–316
access tokens, getting, 99–103
accessing API in background, 362
account management
 checking rate limits, 81–83, 358
 description of, 79–80
 ending user sessions, 83–84
 overview of, 365–368
 updating user notification devices, 85–86
 updating user profile background images, 93–95
 updating user profile colors, 89–91
 updating user profile pictures, 91–93
 updating user profiles, 86–89
 verifying user credentials, 80–81
account methods
 account/end_session, 83–84, 366
 account/rate_limit_status, 68, 81–83, 358, 366
 account/update_delivery_device, 85–86, 366
 account/update_profile, 86–89, 368
 account/update_profile_background_image, 93–95, 367–368
 account/update_profile_colors, 89–91, 366–367
 account/update_profile_image, 91–92, 367
 account/verify_credentials, 80–81, 365
 account/end_session method, 83–84, 366
 account/rate_limit_status method
 description of, 81–83, 366
 to retrieve current rate limit status, 68, 358
accounts. *See also* personal accounts;
 protected accounts; public accounts
 developer, creating, 39–40
 granting permission to applications to access, 74
 private, keeping data in confidential, 360
 for products, 346
 reporting as spam, 176–178
 Twooshes game app and, 300
 user, and @ sign, 8
 account/update_delivery_device method, 85–86, 366
 account/update_profile method, 86–89, 368
 account/update_profile_background_image method, 93–95, 367–368
 account/update_profile_colors method, 89–91, 366–367
 account/update_profile_image method, 91–92, 367
 account/verify_credentials method, 80–81, 365
acquisition of businesses, 343–344
ad management software, 334
ad networks
 cost per thousand, 329–330
 cost per time, 331
 description of, 327–328
 pay per action, 330
 pay per click, 328–329
 vertical, 332–333
addPoint function, 291
addTweet function, 290–291
AdMob ads, 328
advertising. *See also* promotion strategies
 adding to Twitter, 324

- advertising (*continued*)
 - apps for, 58–59
 - paid, to promote applications, 352–353
 - selling, 333–334
 - traditional ad networks, 327–331
 - vertical ad networks, 332–333
 - Agavi framework, 252
 - alt attribute, 18
 - Amazon
 - Payments service, 337
 - SimpleDB data storage system, 319
 - S3 on-demand disk storage space, 256
 - anchor tags, 17
 - angel investors, 326
 - Apache servers, setting up, 258–263
 - API (Application Programming Interface).
 - See also* methods; rate limits; Search API; Search API methods
 - accessing in background, 362
 - authentication, 73–78
 - components of, 65–66
 - defining payload, 71–73
 - description of, 63
 - Google Group for, 358, 362
 - HTTP response status codes and errors, 69–70
 - libraries, 251–252
 - REST, 66, 67–68
 - Streaming, 66
 - terms of service, 63–65
 - tips for, 361–363
 - versioning, 66–67
 - app.ini file, creating, 276–277
 - application definition box, 324
 - application design
 - converting into HTML, 310–311
 - first impressions and, 309
 - hiring professional for, 309–310
 - integrating, 311–313
 - application ecosystem
 - categories in, 47
 - desktop clients, 47–49
 - hardware integration, 60–61
 - mobile clients, 49–50
 - Twitter bots, 59–60
 - Web applications, 50–59
 - application folder, 270
 - Application Programming Interface.
 - See* API
 - APPLICATION_PATH constant, 274
 - applications. *See also* application design; application ecosystem; developing applications; growing pains for applications
 - feature roadmap, 308
 - growth in complexity of, 315
 - licensing and selling, 337–338
 - reasons to develop, 11–12, 242–244
 - reasons to use, 244–246
 - registering with Twitter, 76–77
 - releasing, 308
 - writing first, 41–43
 - array_diff function, 31, 299
 - array_intersect function, 31
 - array_merge function, 31
 - arrays, 30–31
 - array_values function, 31
 - asymmetrical relationship model, 9
 - @replies, 8, 9
 - Austin American Statesmen* (newspaper), 8
 - @AusTraffic, 8
 - authentication, 73–78
 - authenticity in public relations
 - activities, 352
 - auto-follow script, creating, 295–301
 - automating acceptance testing, 315–316
 - avoiding timeouts, 113
- B •**
- background, accessing API in, 362
 - background color, getting from user data, 107–108
 - background style, 23
 - Bailey, Cody Marx (developer), 57
 - BakerTweet wireless device, 61, 395
 - banner ads, 352, 353
 - basic HTTP authentication, 74
 - Beanstalk Web site, 40
 - bids, getting for designers, 310
 - @big_ben_clock, 8
 - blacklisting, 69
 - Blip.fm app, 55, 330, 396
 - block methods
 - blocks/blocking, 172–174, 369
 - blocks/blocking/ids, 174–176, 369

- blocks/create, 166–167, 369–370
 - blocks/destroy, 168–169, 370
 - blocks/exists, 170–172, 370–371
 - overview of, 165–166, 368–369
 - blocking
 - checking for, 170–172
 - getting user details list of blocked users, 172–174
 - retrieving lists of blocked user IDs, 174–176
 - unblocking users, 168–169
 - users, 166–167
 - block-level elements, 16–17
 - blocks/blocking method, 172–174, 369
 - blocks/blocking/ids method, 174–176, 369
 - blocks/create method, 166–167, 369–370
 - blocks/destroy method, 168–169, 370
 - blocks/exists method, 170–172, 370–371
 - Blogger blog platform, 348
 - blogs for product, 348–349
 - body of HTML documents, 16–18
 - bootstrap.php file, creating, 275–276
 - bootstrapping process
 - for applications, 272
 - for self-funding, 325
 - borders, specifying for boxes, 22
 - Botanicals Kit plant monitoring kit, 61, 396
 - box model, 21
 - branches folder, 270
 - brand name. *See also* product
 - building awareness of, 342–343
 - building reputation of, 244
 - monitoring mentions of, 11
 - promoting, 12
 - break tags, closed, 15
 - breaking design HTML into smaller files, 311–313
 - Britcher, Loren (developer), 49
 - Brizzly, 244
 - browsers
 - opening apps in, 42
 - server console inside, 261
 - Brunel, Fred (developer), 59
 - BugMeNot Web site, 271
 - building applications. *See* developing applications
 - Burkert, Ben (developer), 57
 - business issues
 - acquisitions, 343–344
 - building on top of Twitter, 11, 342–344
 - partnering with Twitter, 323
 - venture capital, 323, 325
 - BuySellAds Web site, 331, 353
- C •
- caching, optimizing, 359, 363
 - CafePress Web site, 340
 - CakePHP framework, 252
 - Casale Media ad network, 330
 - Cascading Style Sheets (CSS)
 - basic styles, 21–23
 - description of, 13–14
 - selectors, 20–21
 - SEO and, 353
 - styling HTML with, 18–21
 - case sensitivity of XHTML tags, 15
 - causes, supporting, 12, 244
 - Cawley, Kevin (developer), 50
 - CheapTweet app, 55–56, 397
 - checking. *See* testing
 - class selectors, 20
 - clauses (MySQL), 36
 - click fraud, 328
 - click through rate (CTR), 329
 - Clinkscales, Damon (developer), 54
 - closed break tags, 15
 - cloud computing, 256–257
 - code coverage, 317
 - CodeIgniter framework, 252
 - Coffey, Noah (developer), 363
 - co-location of servers, 256
 - colons in method paths, 116
 - commands (MySQL), 36
 - comments syntax, 25–26
 - Commission Junction ad network, 330
 - communication with users, 360
 - comparing differences between files, 40
 - comparison operators, 27
 - competition, differentiation from, 243–244
 - complexity of software, 315

- conditional statements, 26–28
 - configuration file
 - creating, 276–277
 - defining application environment and, 274
 - constant for directory path, creating, 273–274
 - constants (cURL), 32–33
 - contact management apps, 52–53
 - content, separating from style, 14
 - content tags, 16
 - continuous integration, 317–318
 - controller
 - for auto-follow script, 295–296
 - creating, 279
 - in model-view-control design pattern, 253–254
 - for tweet monitor script, 301–302
 - converting files from PSD to HTML, 310–311
 - Cortesi, Damon (developer), 54, 361
 - cost per thousand (CPM) networks, 329–330
 - cost per time (CPT) networks, 331
 - CoTweet app
 - description of 51
 - as pro version example, 324
 - screenshot, 397
 - as solving problem, 245
 - CouchDB data storage system, 319
 - count function, 31
 - CPM (cost per thousand) networks, 329–330
 - CPT (cost per time) networks, 331
 - Craigslist for design bids, 310
 - create statement, 37
 - created field, 281
 - created_at variable, 287
 - credibility and application design, 309
 - CRM (customer relationship management) apps, 51, 324
 - cron jobs
 - auto-follow script, 295–301
 - description of, 295
 - scheduling, 303
 - tweet monitor script, 301–302
 - CruiseControl continuous integration software, 317–318
 - CSS (Cascading Style Sheets)
 - basic styles, 21–23
 - description of, 13–14
 - selectors, 20–21
 - SEO and, 353
 - styling HTML with, 18–21
 - CTR (click through rate), 329
 - Cummings, Michael (developer), 56
 - cURL
 - description of, 31–33
 - reading HTTP status codes with, 69–70
 - cursor parameter
 - paging through user ids using, 113
 - printing profile pictures using, 109–111
 - customer relationship management (CRM) apps, 51, 324
- D •
- daily trending topics, retrieving, 237–239
 - data fields, naming, 280
 - data models, creating
 - defining tables, 283–284
 - Tweet model, 284–286
 - User model, 286–294
 - data storage, 319
 - data structure, setting up, 280–282
 - Davis, Hayes (developer), 53, 54, 362
 - Davis, Jenn Deering (developer), 54
 - dedicated Web hosting, 256, 257
 - defining application environment, 274
 - degrade gracefully tip, 361
 - delete statement, 38
 - deleting
 - direct messages, 196–198
 - lists, 124–126
 - members from lists, 138–140
 - saved searches, 231–233
 - tweets, 184–186
 - tweets from favorites, 222–224
 - descendant selectors, 20
 - design of application
 - converting into HTML, 310–311
 - first impressions and, 309

- hiring professional for, 309–310
 - integrating, 311–313
 - desktop client applications
 - ad networks and, 328
 - examples of, 47–49
 - selling online, 338
 - details, getting
 - on protected accounts, 109–111
 - on public users, 106–108
 - on specific lists, 122–124
 - develop defensively tip, 361
 - developer accounts, creating, 39–40
 - developer announcement list, 358, 362
 - developer etiquette, 65, 357–360
 - developing applications
 - feature roadmap for, 308
 - ideas for, 241
 - as iterative process, 247
 - libraries and, 251–252
 - model-view-control design pattern and, 253–254
 - motivations for, 11–12, 242–244
 - releasing applications, 308
 - skills for, 246
 - success, defining, 241–242
 - Web application frameworks and, 252–253
 - differentiating product from competition, 243–244
 - direct message methods
 - `direct_messages`, 191–194, 371
 - `direct_messages/destroy`, 196–198, 373
 - `direct_messages/new`, 194–196, 372–373
 - `direct_messages/sent`, 191–194, 372
 - overview of, 190, 371
 - direct messages (DMs)
 - deleting, 196–198
 - description of, 9
 - retrieving, 191–194
 - sending, 194–196
 - `direct_messages` method, 191–194, 371
 - `direct_messages/destroy` method, 196–198, 373
 - `direct_messages/new` method, 194–196, 372–373
 - `direct_messages/sent` method, 191–194, 372
 - directory path, creating constant for, 273–274
 - directory structure, creating, 270–271
 - display style, 23
 - displaying
 - blocked users, 173–174
 - maximum amount of tweets in lists, 127–129
 - rate limits, 318
 - div tags, 17
 - DMs. *See* direct messages
 - doctypes (HTML), 14–15
 - documentation, reading, 358
 - Dodsworth, Iain (developer), 48, 50
 - domain names, setting up, 267–268
 - DOMDocument class, 33–35
 - DOMNodeList object, 34–35
 - donations, asking for, 336–337
 - do-while loops, 20
 - download size, decreasing with external CSS files, 19
 - downloading Zend Framework, 271
- E •
- ease-of-use of applications, 245
 - echo command, 25
 - E lance for design bids, 310
 - Eliason, James (developer), 59
 - elseif parts, 27–28
 - e-mail lists for promoting applications, 349
 - embedding CSS, 19
 - Emma e-mail list manager, 349
 - enabled field, 281
 - enabled variable, 287
 - encoding and OAuth, 78, 97
 - ending user sessions, 83–84
 - enjoyment in building applications, 242
 - entertaining with applications, 245
 - environment, defining, 274
 - `/etc/crontab` file, 303
 - Etzel, Chad (developer), 363
 - Everett, Noah (developer), 54
 - evil, avoiding, tip about, 360

- ExecTweets app, 56, 398
 - Extensible Hypertext Markup Language (XHTML)
 - document example, 15
 - 1.0 Strict, 310, 353
 - Extensible Markup Language (XML)
 - formatting in, 24
 - PHP DOMDocument class and, 33
 - Web browsers and, 42
 - external CSS files, creating, 19
- F •**
- Facebook
 - promoting applications on, 347–348
 - virtual goods sold on, 341, 342
 - FailWhale image, 9
 - family members, as investors, 326
 - Farnham, Aaron (developer), 57
 - favorite methods
 - favorites, 217–220, 373–374
 - favorites/create, 220–222, 374
 - favorites/destroy, 222–224, 374
 - overview of, 217, 373
 - favorites method, 217–220, 373–374
 - favorites/create method, 220–222, 374
 - favorites/destroy method, 222–224, 374
 - feature roadmap, 308
 - Featured Users app
 - description of, 58
 - integrating into applications, 332–333
 - screenshot, 398
 - tweeted testimonials for, 347
 - Twooshes app and, 247
 - files
 - breaking design HTML into smaller, 311–313
 - converting from PSD to HTML, 310–311
 - external CSS, creating, 19
 - rolling back to previous versions, 40
 - uploading to Web servers, 266–267
 - filling needs as reason to build applications, 243
 - Firefox browser and XML, 42
 - Firehose method, 66
 - Fisher, David (developer), 362
 - Flickr
 - product tiers, 339
 - SnapTweet and, 403
 - floating style, 23
 - Follow Cost app, 54, 399
 - FollowcronController.php file, 295
 - \$_followerIds variable, 298
 - followers. *See also* following
 - churning, 64
 - description of, 7
 - getting lists of, 143–145
 - getting user details on, 109–111
 - getting user IDs of, 112–115
 - followers/ids method
 - adding to Zend Framework library, 296–297
 - description of, 112–115, 375
 - following. *See also* followers
 - mass, promotion of, 358
 - users, stopping, 154–155
 - users programmatically, 152–153
 - users to phones, 161–163
 - following lists
 - starting, 145–147
 - stopping, 147–149
 - testing users for, 149–151
 - font styles, specifying, 22–23
 - foreach loops, 28–29
 - form tags, 17
 - formatCreatedAt function, 289
 - fracturing data stores, 319
 - frameset doctype of HTML, 15
 - Francl, Luke (developer), 54
 - freemium model, 339
 - frequency of interaction with system, limits on, 67
 - Friend Or Follow app
 - description of, 11, 52
 - layouts of, 311–312
 - screenshot, 399
 - friends
 - getting user details on, 109–111
 - getting user IDs of, 112–115
 - as investors, 326
 - retweets of, getting, 212–214
 - friends timeline, getting, 201–203
 - friendship methods
 - friendship/create, 152–153, 376
 - friendship/destroy, 154–155, 376–377

- friendships/exists, 156–158, 377
- friendships/show, 158–161, 377–378
- overview of, 151, 376
- friendship/create method,
 - 152–153, 376
- friendship/destroy method, 154–155,
 - 376–377
- friendships/exists method,
 - 156–158, 377
- friendships/show method, 158–161,
 - 377–378
- friends/ids method
 - adding to Zend Framework library,
 - 296–297
 - description of, 112–115, 375
- Front Controller design pattern, 272
- Fuchs, Thomas (developer), 56
- functions
 - addPoint, 291
 - addTweet, 290–291
 - array_diff, 31, 299
 - array_intersect, 31
 - array_merge, 31
 - array_values, 31
 - count, 31
 - cURL, 32
 - formatCreatedAt, 289
 - getFollowerIds, 298
 - getFollowingIds, 298–299
 - getUserTable, 288
 - indexAction, 300–301
 - _initAutoload, 276
 - _initViewHelpers, 276
 - json_decode, 35
 - ksort, 31
 - mapResponse, 288–289
 - PHP, 29–30, 31
 - rawurlencode, 97
 - save, 289
 - subPoint, 291
- fund drives, 337
- funding applications
 - with investors, 326
 - self-funding, 325
- Fung, Yu-Shan (developer), 52

• G •

- games, selling, 341
- geolocation, parameters to facilitate, 182
- GET requests and REST API rate limit,
 - 67–68
- getElementsByTagName method, 34, 35
 - getFollowerIds function, 298
 - getFollowingIds function, 298–299
- getting
 - access tokens, 99–103
 - background colors from user data,
 - 107–108
 - current top 10 trending topics, 235–237
 - details on protected accounts, 109–111
 - details on public users, 106–108
 - details on specific lists, 122–124
 - details on tweets, 180–181
 - direct messages, 191–194
 - favorite tweets of users, 217–220
 - friends timeline, 201–203
 - information about relationships between
 - users, 158–161
 - lists of followers, 143–145
 - members of lists, 134–136
 - protected account details, 109–111
 - public account details, 106–108
 - request tokens, 96–99
 - retweeted status updates, 210–212
 - retweets of friends, 212–214
 - retweets of specific tweets, 188–190,
 - 215–217
 - subscribers to lists, 143–145
 - timelines for lists, 126–129
 - top 20 trending topics per hour, 237–239
 - top 30 trending topics per week, 239–240
 - tweets from public timeline, 199–200
 - tweets that mention screen names,
 - 207–209
 - user details list of blocked users, 172–174
 - user details on followers, 109–111
 - user details on friends, 109–111
 - user IDs of followers, 112–115
 - user IDs of friends, 112–115
 - user timeline, 203–207
- getUserTable function, 288
- GitHub Web site, 40

- giving back to developer community, 359
- GoDaddy service, 267
- Google
 - Ad Manager, 334
 - AdSense, 329
 - AdWords, 352
 - Calendar, updating, 8
 - Checkout service, 337
 - Chrome browser and XML, 42
 - developer announcement list, 358, 362
 - Gmail, 256
 - Group for Twitter API, 358, 362
 - page rank, 354
 - YouTube acquisition, 343
- Gorilla Nation ad network, 330
- graphical banner ads, 352, 353
- growing pains for applications
 - automating acceptance testing and, 315–316
 - continuous integration and, 317–318
 - performance concerns and, 318–319
 - unit testing and, 317

• H •

- Hanson, Keith (developer), 56
- hardware devices, 60–61
- hashtags, 8, 9
- Hashtags.org Web site, 57, 400
- head of HTML documents, 16
- heading tags, 16
- height attribute, 18
- height of element, specifying, 22
- help methods, 378
- help/test method, 378
- Hess, Barry (developer), 54, 361
- hiring designers, 310
- HootSuite app, 51, 324, 400
- horizontal ad networks, 332
- horizontal scaling, 257–258
- hosted version control repositories, 40
- HOT or NOT dating site, 341
- Hoy, Amy (developer), 56
- .htaccess file, creating, 273

- HTML (Hypertext Markup Language)
 - breaking design file into smaller files, 311–313
 - converting files from PSD to, 310–311
 - CSS and, 13–14, 18–21
 - description of, 13
 - elements of, 16–18
 - making valid XHTML, 15
 - SEO and, 353
 - specification documents, 14–15
- HTTP response status codes and errors, 69–70
- Hypertext Markup Language. *See* HTML

• I •

- id parameter, 106, 107
- ID selectors, 20
- ideas
 - going with, 241, 247
 - monetizing, 243
 - motivation and, 242
- if-then-else statements, 26
- image tags, 18
- improving applications, 308
- include_path (PHP), 274
- indexAction function, 300–301
- IndexController.php file
 - creating, 279
 - updating, 303–305
- index.php file, creating, 273–275
- index.php script, 272
- index.phtml file
 - creating, 279
 - updating, 306–307
- industry, networking in, 351
- information aggregation apps, 55–57
- information publishing apps, 57–58
- _initAutoload function, 276
- _initViewHelpers function, 276
- inline elements, 17–18
- insert statement, 38
- installing
 - Apache and PHP on servers, 261–263
 - MySQL on servers, 264–266
 - Zend Framework, 271

integrating
 design HTML into applications, 311–313
 Featured Users into applications, 332–333
 international characters, supporting, 363
 Internet Explorer browser and XML, 42
 investors
 seeking, 326
 venture capital and, 323, 325
 IP addresses
 static, 257
 white listing and, 68
 iPhone App Store, 337, 338
 iterative process, development as, 247

• J •

JASONP support, 363
 JSON (JavaScript Object Notation), 24, 363
 json_decode function, 35

• K •

Kaplan, Philip (developer), 59
 ksort function, 31

• L •

LAMP (Linux, Apache, MySQL, and PHP)
 stack as development platform, 13
 layout template, creating, 277–278
 layout.phtml file
 creating, 278
 HTML layout for, 311
 with references to other files, 312–313
 Le Meur, Loic (developer), 48
 leased servers, 256
 libcurl, 32
 libraries
 OAuth authentication methods, 78
 Twitter API, 251–252
 library folder, 270, 271
 licensing and selling applications, 337–338
 limits. *See* rate limits
 line break tags, 18
 link tags, 16
 links and SEO, 354

LinkShare ad network, 330
 list members methods
 overview of, 134
 :user/:list_id/members, 134–140
 :user/:list_id/members/:id, 140–142
 list methods
 behavior of, 115
 colons and, 116
 :user/lists, 116–118, 120–122
 :user/lists/:id, 118–120, 122–126
 :user/lists/:list_id/statuses,
 126–129
 :user/lists/memberships, 129–131
 :user/lists/subscriptions, 132–134
 list subscribers methods
 overview of, 142–143
 :user/:list_id/subscribers, 143–149
 :user/:list_id/subscribers/:id,
 149–151
 listings
 application/bootstrap.php File, 276
 application/config/app.ini
 File, 277
 application/controllers/
 FollowcronController.php
 File, 295
 application/controllers/
 IndexController.php File, 279
 application/controllers/
 IndexController.php File,
 Updated, 304
 application/controllers/
 TweetcronController.php File,
 301–302
 application/layouts/layout.
 phtml File, 278
 application/models/DbTable/
 Tweet.php File, 284
 application/models/DbTable/
 User.php File, 283
 application/models/Tweet.php File,
 284–285
 application/models/Tweet.
 php:getLastStatusId() File, 285
 application/models/User.php
 File, 287
 application/models/User.
 php:addPoint() and subPoint()
 File, 291

listings (*continued*)

application/models/User.
php:addTweet() File, 290–291

application/models/User.
php:constructor File, 288

application/models/User.
php:mapResponse() File, 289

application/models/User.
php:save() File, 290

applications/views/scripts/
index/index.phtml File, 306

application/views/scripts/
index.phtml File, 279

application/views/scripts/
pagination_control.phtml File,
305–306

array_dif Function, 299

Authentication Login URL, 96

Background Styles, 23

Blocking Users, 167

Border Styles, 22

Check if Two Users Follow Each Other,
160–161

Check if User Is Blocked, 171–172

Check Length of DOMNodeList, 35

Concatenated oauth_signature String
Before Encrypting and Parameter
Encoding, 97, 101

Count and List of Blocked User IDs,
Getting, 175–176

Create Statement, 37

Creating Array, 30

Creating DOMDocument Object, 34

Current Trends, Retrieving, 236–237

Cursor, Using to Print Profile Pictures,
110–111

Daily Trending Topics, Retrieving,
238–239

Delete Statement, 38

Deleting Direct Message, 197–198

Deleting Tweet, 185–186

Display Style, 23

Displaying Blocked Users, 173–174

do-while Loop that Counts to 10, 20

Elseif Part, 27–28

Ending Persistent Basic HTTP Auth
Session, 84

Error Message in XML, 70
/etc/crontab File, 303

Favorites, Adding Tweet to, 221–222

Favorites, Removing Tweet from, 223–224

Floating Style, 23

Following New Users, 153

Font Decorations, 23

foreach Loop, 28, 29

Function example, 30

Generating Access Token, 101–103

Generating Request Token, 98–99

getElementsByTagName, 34, 35

getFollowerIds Function, 298

getFollowingIds Function, 299

Go Straight to First Node in DOMNodeList
Object, 34

Height and Width, 22

HelloTwitter.html, 41

If-Then-Else, 26

indexAction Function, 300–301

index.php File, Beginning of, 273

index.php File, Complete, 275

index.php File, Defining Application
Environment in, 274

Insert Statement, 38

JSON Object example, 24

Last 100 Retweets of Friends, Getting,
213–214

library/Zend/Service/Twitter.
php Directory, Adding Social Graph
Methods, 296–297

List, Creating, 117–118

List, Deleting, 125–126

List, Updating, 119–120

List Details, Getting, 123–124

List Followers, Getting, 144–145

List Members, Adding, 137–138

List Members, Removing, 139–140

List Membership, Checking, 141–142

List Subscribers, Adding, 146–147

List Tweets, Getting, 128–129

Lists User Belongs To, Getting, 130–131

Lists User Subscribes To, Getting,
132–134

Loop through DOMNodeList Object, 34

Margin and Padding, 21

Margin and Padding Shorthand, 22

Marking Users as Spam, 177–178

Most Recent 100 Retweets, Getting,
211–212

- Most Recent 100 Retweets of Tweets, Getting, 216–217
- oauth/access_token Response Body, 100
- oauth/request_token Response Body, 96
- Paging Through Friends Timeline, 202–203
- Paging Through Tweets that Mention Screen Name, 208–209
- Paging Through User Timeline, 205–207
- PHP Comments Syntax, 26
- Post Tweet (curl command line client), 32
- Post Tweet (libcurl), 33
- Post Tweet (`statuses/update` method), 183–184
- Previous Tweets, Retrieving and Displaying, 219–220
- Print String to Screen, 25
- Profile Colors, Updating, 90–91
- Profile Pictures of Most Recent 20 Public Tweets, Printing, 200
- Pull Up Public Timeline, 32
- Rate Limit Status, Looking Up, 82–83
- Reading HTTP Status Code with PHP and cURL, 69
- Received Direct Messages, Getting Using Paging, 192–194
- Receiving Notifications from Users, 162–163
- Retweeting Tweets, 187–188
- Retweets of Specific Tweets, Retrieving, 189–190
- Saved Search Details, Getting, 228–229
- Saved Searches, Creating, 230–231
- Saved Searches, Removing, 232–233
- Saved Searches, Retrieving All, 226–227
- Searching For Tweets Using API, 235
- Select Query Using Group By and Having Clause, 37
- Select Query Using Where, Order By, and Limit Clause, 37
- Sending Direct Message, 195–196
- SQL Select Query, 36
- SQL Select Query Using Join, 37
- `status` Object in XML, 72–73
- Subscription Status, Checking, 150–151
- Testing if One User Follows Another, 157–158
- Turning Off SMS Notifications, 164–165
- Tweet Details, Getting, 180–181
- Tweet Table, Creating, 282
- `Tweet.php` File, 286
- Twitter Status Update XML Response, 42–43
- Twooshes `layout.phtml` File, 311
- Twooshes `layout.phtml` File With References to Other Files, 312–313
- Unblocking Users, 168–169
- Unfollowing Lists, 148–149
- Unfollowing Users, 154–155
- Update Statement, 38
- Updating Device Settings, 85–86
- Updating Twitter Background, 93–95
- Updating Twitter Picture, 92–93
- Updating Twitter Profile, 87–89
- User Agent String to Increase Search API Rate Limit, 68
- User Data, Getting with `users/show` Method, 107–108
- User IDs of Mutual Followers, Getting, 114–115
- User Is Not Blocked Message, 170
- `user` Object in XML example, 71–72
- User Table, Creating, 280–281
- Username and Password, Checking, 81
- `User.php` File, 292–294
- Users List Follows, Getting, 135–136
- Users List of Lists, Getting, 121–122
- Weekly Trending Topics, Retrieving, 240
- `www/.htaccess` File, Creating, 273
- XML Object, 24
- XML Response from `friendships/show` Method, 158–159
- `Zend_Application` Object, Creating and Running Bootstrap, 275
- `Zend_Service_Twitter` Object, Creating, 298
- lists
 - adding members to, 136–138
 - of blocked user IDs, getting, 174–176
 - of blocked users, getting, 172–174
 - creating new, 116–118
 - deleting, 124–126
 - deleting members from, 138–140
 - description of, 115
 - details on specific lists, getting, 122–124
 - following, 145–147

lists (*continued*)

- members of, getting, 134–136
- stop following, 147–149
- subscribers to, getting, 143–145
- testing if users are members of, 140–142
- timelines for, getting, 126–129
- updating existing, 118–120
- user belongs to, getting, 129–131
- user follows, getting, 132–134
- user list of lists, getting, 120–122
- local cache of detailed user data, 114
- logging in
 - authentication, 73–78
 - users with OAuth, 95–96
- logical operators, 27
- loops, 28–29
- Loving, Adam (developer), 363

• M •

Magpie ad network

- description of, 58, 324, 332
- screenshot, 401
- Twooshes app and, 247
- MailChump e-mail list manager, 349
- managed servers, 256
- mapResponse function, 288–289
- margins, specifying for boxes, 21–22
- Mashable Web site, 351
- mass following, promotion of, 358
- max_id parameter, 192
- media sharing apps, 54–55
- members
 - adding to lists, 136–138
 - deleting from lists, 138–140
 - of lists, getting, 134–136
 - testing if users are list members, 140–142
- Menti, Mario (developer), 58
- mentions
 - of brand name, monitoring, 11
 - description of, 9
 - of screen name, getting, 207–209
- meta tags, 16
- methods
 - account management, 79–91, 365–368
 - block, 165–176, 368–371
 - colons in method paths, 116
 - direct messages, 190–198, 371–373

- favorite, 217–224, 373–374
- friendship, 151–161, 376–378
- help, 378
- list, 115–134
- list members, 134–142
- list subscribers, 142–151
- notification, 161–165, 378–379
- OAuth authentication, 78, 95–103, 379–380
- saved searches, 224–233, 380–382
- Search API, 233–240, 382–384
- social graph, 112–115, 296–297, 374–375
- spam reporting, 176–178, 384–385
- status, 179–190, 385–394
- timeline, 198–217
- user, 105–111, 394
- version numbers of, 67
- for working with OAuth, 75
- mobile client applications
 - ad networks and, 328
 - examples of, 49–50
- mobile providers, SMS revenue sharing
 - from, 324
- model-view-control design pattern, 253–254. *See also* controller; view
- mod_rewrite, installing, 262–263
- money, making. *See also* advertising
 - asking for donations, 336–337
 - building businesses, 342–344
 - developing apps, 12
 - as reason to build applications, 242–243
 - requesting payment for service, 335–339
 - revenue model speculations regarding
 - Twitter, 324
 - revenue models, 337–338
 - selling goods, 339–342
 - selling subscriptions, 338–339
- monitoring
 - bakeries for fresh items, 61, 395
 - mentions of brand name or product, 11
 - plants for water needs, 61, 396
 - trending words, 10, 236–240
- Morse, Jacob (developer), 56
- motivations for building applications, 11–12, 242–244
- Movable Type blog platform, 348
- Mr. Tweet app, 52, 401

MVC (model-view-control) design pattern, 253–254. *See also* controller; view

My Cloud Servers dashboard, 259

my.cnf file (MySQL), 265, 266

MySQL

description of, 35

my.cnf file, 265, 266

servers, setting up, 263–266

• N •

name recognition, building, 342–343. *See also* brand name

name squatting, 64

naming data fields, 280

Nearly Free Speech Web hosting

company, 267

networking in industry, 351

news organizations, updates from, 10

next_cursor node, 109–110

non-relational data storage systems, 319

notification methods

notifications/follow, 161–163, 378–379

notifications/leave, 163–165, 379

overview of, 161, 378

notifications/follow method, 161–163, 378–379

notifications/leave method, 163–165, 379

numeric userIDs, 107

• O •

OAuth

access tokens, getting, 99–103

description of, 74–78

methods, 379

parameter encoding with, 78, 97

request tokens, getting, 96–99

using, 359–360

oauth/access_token method, 99–103, 380

oauth/authenticate method, 75, 95–96, 380

oauth/authorize method, 75, 95–96, 380

oauth/request_token method, 96–99, 380

Odeo podcasting company, 325

ON DUPLICATE KEY UPDATE command, 289–290

oneforty Web site, 354

140tees Web site, 340

online payments, accepting, 337

open API, 11

Open Source Web Design Web site, 310

OpenX ad management software, 334

opt-in e-mail lists, 349

output, 71–73

• P •

padding, specifying for boxes, 21–22

pagination_control.phtml file, 305–306

paragraph tags, 16

parameter encoding with OAuth, 78, 97

parameters to facilitate geolocation, 182

parsing standard DOM markup, 33

partnering with Twitter, 323

passwords

root, using to connect applications to databases, 265

of users, checking, 80–81

of users, handling, 74

pay per action (PPA) networks, 330

pay per click (PPC) networks, 328–329

payload, defining, 71–73

payment for services, requesting, 335–339

PayPal service, 337

performance concerns, 318–319

permission

asking before acting on user's behalf, 357

granting to applications to access accounts, 74

Perrin, Andrew (developer), 363

personal accounts

allowing access to, 41

for development, creating, 39–40

experience with, 39

Philips, Trey (developer), 56

phones, following users to, 161–163

photo-sharing sites, 408

PHP

- arrays, 30–31
- comments syntax, 25–26
- conditional statements, 26–28
- cURL, 31–33
- description of, 25
- DOMDocument class, 33–35
- echo command, 25
- frameworks, 252
- functions, 29–30
- includes, 313
- json_decode function, 35
- loops, 28–29
- online manual, 31
- rawurlencode function, 97
- reading HTTP status codes with, 69–70
- servers, setting up, 258–263
- PHP Zend Framework
 - adding methods to library, 296–297
 - bootstrap process, 272
 - bootstrap.php file, creating, 275–276
 - configuration file, creating, 276–277
 - description of, 252–253
 - directory structure, creating, 270–271
 - include_path, setting to library, 274
 - installing, 271
 - layout template, creating, 277–278
 - mod_rewrite, installing, 262–263
 - scoreboard, creating, 303–307
 - setting up, 269–279
 - Table Data Gateway design pattern, 283
 - testing, 279
 - unit tests and, 317
 - view and controller, creating, 278–279
- PHPUnit unit-testing framework, 317
- physical goods, selling, 339–340
- plant monitoring kits, 61, 396
- platforms for blogs, 348
- popular applications, imagining and building, 241–242
- power consumption, tracking, 61, 404
- PPA (pay per action) networks, 330
- PPC (pay per click) networks, 328–329
- previous_cursor node, 109–110
- printing profile pictures
 - of most recent 20 public tweets, 200
 - of people followed, 110–111

- private accounts, keeping data in
 - confidential, 360
- problem solving with applications, 245
- product. *See also* brand name
 - blogs for, 348–349
 - differentiating from competition, 243–244
 - Facebook fan pages for, 348
 - monitoring mentions of, 11
 - Twitter accounts for, 346
- profile_image_url variable, 287
- profiles
 - background images, updating, 93–95
 - colors, updating, 89–91
 - data in, 79
 - pictures, updating, 91–93
 - updating, 86–89
- Project Wonderful Web site, 353
- promotion strategies. *See also* advertising
 - blogs, 348–349
 - on Facebook, 347–348
 - opt-in e-mail lists, 349
 - overview of, 345
 - public relations, 351–352
 - search engine optimization, 353–354
 - social networking, 346–349
 - on Twitter, 346–347
 - viral marketing, 349–350
- protected accounts
 - following, 152–153
 - getting details of, 109–111
- PSD to HTML file conversion service, 311
- public accounts
 - asymmetrical relationship model and, 9
 - getting details of, 106–108
- public relations strategies, 351–352
- public timeline
 - getting tweets from, 199–200
 - Search API and, 233



- quotation marks and XHTML
 - tag attributes, 15

• R •

- Rackspace Cloud Servers
 - Apache and PHP, setting up, 258–263
 - description of, 257
 - domain names, setting up, 267–268
 - MySQL, setting up, 263–266
 - uploading files to, 266–267
- rate limits
 - blacklisting and, 69
 - checking, 81–83
 - displaying, 318
 - getting around, 363
 - REST API, 67–68
 - Search API, 68–69, 233
 - staying within, 358
 - storing API results locally and, 107
- `rawurlencode` function, 97
- reading documentation, 358
- Reagan, Dusty (developer), 52, 58
- receiving notifications from users, 162–163
- registering applications, 76–77
- registration page, 76, 77
- relational databases, 36, 319
- relationship between users, getting
 - information about, 158–161
- relationship structure, 9
- releasing applications, 308
- Remember The Milk app, 60, 402
- removing
 - direct messages, 196–198
 - lists, 124–126
 - members from lists, 138–140
 - saved searches, 231–233
 - tweets, 184–186
 - tweets from favorites, 222–224
- reporting tools, 360
- `report_spam` method, 176–178, 384–385
- Representational State Transfer (REST), 63
- reputation, building by developing
 - applications, 12, 244
- request tokens, getting, 96–99
- resources
 - for building applications, 246
 - online, 65
- REST (Representational State Transfer), 63
 - REST API
 - description of, 66
 - rate limit, 67–68
 - result set, 71–73
 - retrieving
 - access tokens, 99–103
 - background colors from user data, 107–108
 - current top 10 trending topics, 235–237
 - details on protected accounts, 109–111
 - details on public users, 106–108
 - details on specific lists, 122–124
 - details on tweets, 180–181
 - direct messages, 191–194
 - favorite tweets of users, 217–220
 - friends timeline, 201–203
 - information about relationships between users, 158–161
 - lists of followers, 143–145
 - members of lists, 134–136
 - protected account details, 109–111
 - public account details, 106–108
 - request tokens, 96–99
 - retweeted status updates, 210–212
 - retweets of friends, 212–214
 - retweets of specific tweets, 188–190, 215–217
 - subscribers to lists, 143–145
 - timelines for lists, 126–129
 - top 20 trending topics per hour, 237–239
 - top 30 trending topics per week, 239–240
 - tweets from public timeline, 199–200
 - tweets that mention screen names, 207–209
 - user details list of blocked users, 172–174
 - user details on followers, 109–111
 - user details on friends, 109–111
 - user IDs of followers, 112–115
 - user IDs of friends, 112–115
 - user timeline, 203–207
 - retweeting tweets, 186–188
 - retweets (RTs)
 - description of, 8, 9
 - of specific tweets, retrieving, 188–190, 215–217
 - reusing markup across multiple layouts, 313

- revenue model speculations about
 - Twitter, 324
 - revenue models
 - asking for donations, 336–337
 - building businesses, 342–344
 - licensing and selling software, 337–338
 - requesting payment for services, 335–339
 - selling goods, 339–342
 - selling subscriptions, 338–339
 - Right Banners service, 353
 - Roelands, Duane (developer), 362
 - rolling back files to previous versions, 40
 - root usernames and passwords, using to
 - connect applications to databases, 265
 - Rose, Kevin (developer), 52
 - RTs. *See* retweets
 - rules
 - for developers, 65
 - general, 64
- S •**
- SaaS (software as a service) view, 338
 - Safari browser and XML, 42
 - save function, 289
 - saved searches
 - creating, 229–231
 - details, getting, 227–229
 - removing, 231–233
 - retrieving all, 225–227
 - saved searches methods
 - overview of, 224–225, 380
 - saved_searches, 225–227, 381
 - saved_searches/create, 229–231, 381
 - saved_searches/destroy, 231–233, 381
 - saved_searches/show, 227–229, 381–382
 - saved_searches method, 225–227, 381
 - saved_searches/create method, 229–231, 381
 - saved_searches/destroy method, 231–233, 381
 - saved_searches/show method, 227–229, 381–382
 - saving User object to database, 289
 - scalability of Web applications, 318–319
 - scheduling cron jobs, 303
 - score field, 281
 - scoreboard for Twooshes game, creating, 303–307
 - screen names
 - @ replies and, 8
 - tweets that mention, getting, 207–209
 - screen_name parameter, 107, 362
 - screen_name variable, 287
 - script tags, 16
 - scripts folder, 271
 - Search API
 - description of, 66
 - rate limit, 68–69
 - Search API methods
 - overview of, 233, 382
 - search.tweet.com/search, 233–235, 382–383
 - search.twitter.com/trends, 235–237, 383
 - search.twitter.com/trends/current, 235–237, 384
 - search.twitter.com/trends/daily, 237–239, 383–384
 - search.twitter.com/trends/weekly, 239–240, 384
 - search engine optimization (SEO), 348, 353–354
 - search functionality, 11
 - searching for tweets, 233–235
 - search.twitter.com/search method, 233–235, 382–383
 - search.twitter.com/trends method, 235–237, 383
 - search.twitter.com/trends/current method, 235–237, 384
 - search.twitter.com/trends/daily method, 237–239, 383–384
 - search.twitter.com/trends/weekly method, 239–240, 384
 - SecretTweet app, 57, 402
 - Secure Sockets Layer (SSL), 246
 - securing MySQL server, 265–266
 - Seismic Desktop app, 48, 403
 - select query, 36
 - selectors (CSS), 20
 - Selenium, acceptance testing with, 316
 - self-funding applications, 325

- selling
 - advertising, 333–334
 - applications, 337–338
 - goods, 339–342
 - subscriptions, 338–339
- sending
 - direct messages, 194–196
 - status updates, 183–184
- SEO (search engine optimization), 348, 353–354
- server console inside Web browsers, 261
- Server Side Includes (SSI), 313
- servers
 - Apache and PHP, setting up, 258–263
 - domain names, setting up, 267–268
 - hosting providers, selecting, 257–258
 - increasing performance of, 319
 - MySQL, setting up, 263–266
 - splitting workload across, 257–258
 - uploading files to, 266–267
 - Web hosting solutions, 255–257
- shared Web hosting, 255, 257
- shareware model, 338
- Short Message Service (SMS) revenue
 - sharing from mobile providers, 324
- 64-bit integers, using, 362
- Smith, Brian (developer), 57
- Smith, Kevin (developer), 57
- SMS (Short Message Service) revenue
 - sharing from mobile providers, 324
- SMS updates
 - turning off, 164–165
 - usefulness of, 10
- SnapTweet app, 55, 403
- social capital, building, 342–343
- social graph methods
 - adding to Zend Framework library, 296–297
 - overview of, 374–375
 - user IDs of friends and followers, getting, 112–115
- social networking sites
 - promotion strategies using, 346–349
 - virtual goods sold on, 341–342
- software. *See* applications
- software as a service (SaaS) view, 338
- software frameworks, 252–253
- solving problems with applications, 245
- source code, viewing, 42
- spam reporting method, 176–178, 384–385
- splitting workload across servers, 257–258
- SSI (Server Side Includes), 313
- SSL (Secure Sockets Layer), 246
- static IP addresses, 257
- statistical analysis apps, 53
- status methods
 - overview of, 179, 385
 - statuses/destroy, 184–186, 386
 - statuses/retweet, 186–188, 390
 - statuses/retweets, 188–190, 392
 - statuses/show, 180–181, 392
 - statuses/update, 182–184, 392–393
- status objects, 71, 72–73
- status updates, retweeted, getting, 210–212
- statuses/destroy method, 184–186, 386
- statuses/followers method, 109–111, 386–387
- statuses/friends method, 109–111, 387
- statuses/friends_timeline method, 201–203, 388
- statuses/home_timeline method, 388–389
- statuses/mentions method, 207–209, 389
- statuses/public_timeline method, 199–200, 389
- statuses/retweet method, 186–188, 390
- statuses/retweeted_by_me method, 210–212, 390
- statuses/retweeted_to_me method, 212–214, 391
- statuses/retweets method, 188–190, 392
- statuses/retweets_of_me method, 215–217, 390–391
- statuses/show method, 180–181, 392
- statuses/update method, 182–184, 392–393
- statuses/user_timeline method, 203–207, 393–394

- Stone, Biz (founder), 324
 - stop following
 - lists, 147–149
 - users, 154–155
 - stop receiving notifications from users, 163–165
 - storing API results locally, 107, 280
 - Streaming API, 66
 - “strict” doctype of HTML, 15, 310, 353
 - strong tags, 18
 - style, separating from content, 14
 - style attribute, 18
 - style tags, 16, 19
 - subPoint function, 291
 - subscribers to lists
 - adding, 146–147
 - getting, 143–145
 - testing for, 149–151
 - subscribing to Google developer
 - announcement list, 358, 362
 - subscriptions, selling, 338–339
 - Subversion version control system, 40, 317
 - success, definition of, 241
 - Summize acquisition, 343–344
 - Symfony framework, 252
- T •**
- Table Data Gateway design pattern, 283
 - table tags, 17
 - tables, creating in MySQL, 37
 - tags directory, 270
 - TechCrunch Web site, 351
 - terminology, 9
 - terms of service
 - developer etiquette, 65
 - general, 64
 - testing
 - automating acceptance testing, 315–316
 - if users are list members, 140–142
 - if users follow lists, 149–151
 - if users follow users, 156–158
 - unit, 317
 - users for blocking, 170–172
 - Zend Framework, 279
 - text ads, 352
 - text parameter, 195
 - textual data formatting, 24
 - timeline methods
 - overview of, 198–199
 - statuses/friends_timeline, 201–203, 388
 - statuses/mentions, 207–209, 389
 - statuses/public_timeline, 199–200, 389
 - statuses/retweeted_by_me, 210–212, 390
 - statuses/retweeted_to_me, 212–214, 392
 - statuses/retweets_of_me, 215–217, 390–391
 - statuses/user_timeline, 203–207, 393–394
 - timelines
 - for lists, getting, 126–129
 - public, 199–200, 233
 - timeouts, avoiding, 113
 - Tiny Twitter app, 50, 404
 - title attribute, 18
 - title tags, 16
 - tooting own horn, 351
 - Torrone, Phil (developer), 61
 - tracking
 - ads sold, 334
 - file changes over time, 40
 - power consumption, 61, 404
 - trademarks, 359
 - traffic reports, accessing, 8
 - traits of respectable developers, 357–360
 - transitional doctype of HTML, 15
 - trending topics
 - current top 10, getting, 235–237
 - monitoring, 10
 - top 20 per hour, getting, 237–239
 - top 30 per week, retrieving, 239–240
 - trendingTopics field, 281
 - Tribal Fusion ad network, 329
 - trunk folder, 270
 - trust, building with users, 246, 346
 - turning off SMS updates, 164–165
 - Tweet Congress application, 244

- tweet database table, building, 282
 - Tweet model, creating, 284–286
 - Tweet-a-Watt device, 61, 404
 - TweetcronController.php file, 301–302
 - TweetDeck app
 - for desktops, 48
 - managing API limits with, 318
 - for mobile phones, 50
 - screenshot, 405
 - Tweetie app, 49–50, 337
 - Tweeting Too Hard app, 56, 405
 - Tweetmeme app, 349–350
 - Tweetname bot, 60, 406
 - Tweet.php file, 284–286
 - TweetReach app, 53, 406
 - tweets. *See also* timeline methods
 - adding to favorites, 220–222
 - creating, 182–184
 - deleting, 184–186
 - description of, 7, 9
 - details of, getting, 180–181
 - favorite, retrieving, 217–220
 - in lists, displaying maximum amount of, 127–129
 - mentioning screen name, getting, 207–209
 - from public timeline, getting, 199–200
 - purposes of, 7–8
 - removing from favorites, 222–224
 - retrieving retweets of, 188–190, 215–217
 - retweeting, 186–188
 - searching for, 233–235
 - TweetStats app, 54, 245, 407
 - Twibes, 363
 - Twistori app, 56–57, 407
 - TwitCause application, 244
 - Twitdom Web site, 354
 - TwitPic app, 54–55, 408
 - Twittad app, 59, 408
 - Twitter
 - application definition box, 324
 - bots, 59–60
 - building businesses on top of, 11, 342–344
 - description of, 7
 - features of, 8, 9–11
 - open API, 11
 - partnering with, 323
 - promoting applications on, 346–347
 - relationship structure of, 9
 - revenue model speculations, 324
 - SMS updates, 10
 - Summize acquisition, 343–344
 - terminology of, 9
 - trends and searches, 10–11
 - wiki, 78, 252
 - Twitter-async library, 78
 - Twittercal bot, 59, 409
 - TwitterCounter app, 53, 409
 - Twitterfeed app, 57–58, 410
 - twitter_id field, 280
 - twitter_id variable, 287
 - Twitterrific for Mac app
 - description of, 49, 338
 - logo T-shirts, 340
 - screenshot, 410
 - twooshes field, 281
 - Twooshes game app
 - auto-follow script, creating, 295–301
 - bootstrap.php file, creating, 275–276
 - bootstrapping, 272
 - cloud hosting solution for, 257
 - configuration file, creating, 276–277
 - cron jobs, creating, 295–303
 - data models, creating, 283–294
 - data structure, setting up, 280–282
 - description of, 9, 247
 - directory structure, creating, 270–271
 - .htaccess file, creating, 273
 - index.php file, creating, 273–275
 - layout template, creating, 277–278
 - scoreboard, creating, 303–307
 - view and controller, creating, 278–279
 - Web server dashboard, 260
 - Zend Framework, installing, 271
 - Twtbase Web site, 354
 - type selectors, 20
- u •
- unblocking users, 168–169
 - unfollowing
 - lists, 147–149
 - users, 154–155

- Unfuddle Web site, 40
- unit testing, 317
- unordered list tags, 16
- update statement, 38
- updated field, 281
- updates, following and posting, 10
- updating
 - existing lists, 118–120
 - Google Calendar, 8
 - user notification devices, 85–86
 - user profile background images, 93–95
 - user profile colors, 89–91
 - user profile pictures, 91–93
 - user profiles, 86–89
- uploading files to Web servers, 266–267
- usage quotas, getting around, 363
- user accounts and @ sign, 8
- User Agent string to increase Search API rate limit, 68–69
- user credentials, verifying, 80–81
- user database table, building, 280–281
- user details list
 - of blocked users, getting, 172–174
 - of friends and followers, getting, 109–111
- user IDs
 - of friends and followers, getting, 112–115
 - list of blocked, getting, 174–176
- user methods
 - details of friends and followers, getting, 109–111
 - details of users, getting, 106–108
 - overview of, 105, 394
- User model, creating, 286–294
- user notification devices, updating, 85–86
- user objects, 71–72
- user passwords, handling, 74
- user profiles
 - background images, updating, 93–95
 - colors, updating, 89–91
 - data in, 79
 - pictures, updating, 91–93
 - updating, 86–89
- user sessions, ending, 83–84
- user timeline, getting, 203–207
- userId field, 281
- user_id parameter, 107, 362
- :user/:list_id/members method
 - adding members to lists, 136–138
 - getting lists of members, 134–136
 - removing members from lists, 138–140
- :user/:list_id/members/:id method, 140–142
- :user/:list_id/subscribers method
 - following lists, 145–147
 - getting list of followers, 143–145
 - stop following lists, 147–149
- :user/:list_id/subscribers/:id method, 149–151
- :user/lists method, 116–118, 120–122
- :user/lists/:id method, 118–120, 122–126
- :user/lists/:list_id/statuses method, 126–129
- :user/lists/memberships method, 129–131
- :user/lists/subscriptions method, 132–134
- usernames
 - checking, 80–81
 - root, using to connect applications to databases, 265
- User.php file
 - complete syntax, 292–294
 - creating, 283
 - starting class and defining fields, 287
- users
 - blocking, 166–167
 - building trust with, 246, 346
 - checking for blocking, 170–172
 - communication with, 360
 - favorite tweets of, retrieving, 217–220
 - following, 152–153
 - getting information about relationship between, 158–161
 - list of lists, getting, 120–122
 - receiving notifications from, 161–163
 - stop following, 154–155
 - stopping notifications from, 163–165
 - unblocking, 168–169
- users/show method, 106–108, 394

• U •

validating Web sites as XHTML 1.0 Strict, 353–354

ValueClick ad network, 330

van Zanten, Boris Veldhuijzen (developer), 53

Venneman, Brian (developer), 54

venture capital

- description of, 323, 325
- seeking, 326

verifying user credentials, 80–81

version control, 40, 317

versioning of API, 66–67

vertical ad networks, 332–333

view

- for auto-follow script, 296
- creating, 278–279
- in model-view-control design pattern, 253–254

viewing source code, 42

viral marketing, 349–350

virtual goods, selling, 339, 341–342

• W •

Web application frameworks

- breaking up HTML markup and, 313
- description of, 252–253

Web applications

- advertising, 58–59
- contact management, 52–53
- customer relationship management, 51
- description of, 50–51
- information aggregation, 55–57
- information publishing, 57–58
- media sharing, 54–55
- selling subscriptions for, 338–339
- statistical analysis, 53

Web banners, 352, 353

Web browsers

- opening apps in, 42
- server console inside, 261

Web development. *See also* CSS; HTML; MySQL; PHP

LAMP (Linux, Apache, MySQL, and PHP) stack and, 13

- separating content from style in, 14
- standards for, 14

Web hosting companies, 257–258

Web hosting solutions

- cloud computing, 256–257
- dedicated, 256
- shared, 255

Web servers

- Apache and PHP, setting up, 258–263
- domain names, setting up, 267–268
- hosting providers, selecting, 257–258
- increasing performance of, 319
- MySQL, setting up, 263–266
- splitting workload across, 257–258
- uploading files to, 266–267
- Web hosting solutions, 255–257

Web sites

- ad networks, 329–330, 331, 332, 353
- AdMob, 328
- advertising apps, 58–59
- Brizzly, 244
- BugMeNot, 271
- BuySellAds, 331
- CafePress, 340
- cloud computing examples, 256
- contact management apps, 52–53
- CoTweet, 245
- CRM apps, 51
- CruiseControl, 318
- for design bids, 310
- desktop clients, 48–49
- developer announcement list, 362
- for domain names, 267
- e-mail list managers, 349
- Facebook, 341, 342
- Featured Users, 247, 332–333
- hardware devices, 61
- hosted version control repositories, 40
- HOT or NOT, 341
- information aggregation apps, 55–57
- information publishing apps, 57–58
- lists and reviews, 354
- Magpie ad network, 247, 332
- media sharing applications, 54–55
- mobile clients, 50

Web sites (*continued*)

- non-relational data storage systems, 319
 - 140tees, 340
- Open Source Web Design, 310
- PHP frameworks, 252
- PHP manual, 31
- PSD to HTML file conversion service, 311
- Rackspace Cloud Servers, 257
- reporting tools, 360
- resources, 65
- Right Banners, 353
- statistical analysis apps, 53
- Streaming API, 66
- Subversion, 317
- tech writers, 351
- Tribal Fusion ad network, 329
- Tweet Congress, 244
- TweetDeck, 318
- TweetStats, 245
- TwitCause, 244
- Twitter bots, 59–60
- Twitter wiki, 78, 252
- validating as XHTML 1.0 Strict, 353–354
- white listing application, 68, 107
 - `Zend_Paginator` class, 307
- WebFollow app, 52–53
- website directory, 270
- weekly trending topics, retrieving, 239–240
- WeFollow app, 350, 411
- weight attribute, 18
- white listing, 68, 107
- width of element, specifying, 22
- Wikipedia donation page, 336–337
- Williams, Abraham (developer), 362
- Williams, Evan (founder), 324
- Wilson, Mike (@2drinksbehind), 10
- Wordpress blog platform, 348
- workflow of model-view-control design
 - pattern, 253–254
- workload, splitting across servers, 257–258
- World Wide Web Consortium (W3C), 14
- writing first app, 41–43
- W3C (World Wide Web Consortium), 14
- W3C Markup Validation Service, 353–354
- www folder, 271

• X •

- XHTML (Extensible Hypertext Markup Language)
 - document example, 15
 - 1.0 Strict, 310, 353
- XML (Extensible Markup Language)
 - formatting in, 24
 - PHP `DOMDocument` class and, 33
 - Web browsers and, 42
 - `xmlns` attribute, 15

• Y •

- Yasuda, Jeff (developer), 54
- YouTube, acquisition of, 343

• Z •

- Zend Framework
 - adding methods to library, 296–297
 - bootstrap process, 272
 - `bootstrap.php` file, creating, 275–276
 - configuration file, creating, 276–277
 - description of, 252–253
 - directory structure, creating, 270–271
 - `include_path`, setting to library, 274
 - installing, 271
 - layout template, creating, 277–278
 - `mod_rewrite`, installing, 262–263
 - scoreboard, creating, 303–307
 - setting up, 269–279
 - Table Data Gateway design pattern, 283
 - testing, 279
 - unit tests and, 317
 - view and controller, creating, 278–279
- `Zend_Application` object, creating, 274–275
- `Zend_Paginator` class, 303, 304, 307
- `Zend_Service_Twitter` object,
 - creating, 297–298