

Index

■ Numbers & Symbols

`$()` function (Prototype library), 133
`@SuppressWarnings` annotation, 321

■ A

Accelerometer service (Mojo), 102–103
account sign up (GAE), 311
account verification (Twitter application), 300–301
`account/verify_credentials` API (Twitter), 275–276
Accounts service (Mojo), 92
`activate()` method (Mojo), 27, 148
activating

- and deactivating Details scene (business search application), 212–214
- gameScreen scene (Engineer game), 251–253

`activeScene()` method, 127
activities feature (webOS), 15–16
Adaptive Path, 168
Add Category scene (Code Cabinet application), 137
Address objects, 96
`advanceFocus()` method (Mojo.view), 65
AJAX (Asynchronous JavaScript and XML)

- `Ajax.Request()` method, 205
- overview, 167–168
- same-domain policy and, 168–169

Alarms service (Mojo), 92–93
Animation (Mojo), 46–47
annotations, Java, 317
APIs

- mappings (REST), 315–317
- Mojo framework, 22

overview (Mojo), 43–45
services, 169
Twitter, 2284
Yahoo! Maps, 172
App Catalog (Palm), 26–27
app controller (Mojo), 52–53
App Store, 9–10
`AppAssistant()` function, 282
appinfo.json file

- application ID stored in, 38–40
- business search application, 175–176
- Code Cabinet application, 113
- defined, 23
- Engineer game application, 232–233
- Twitter application, 279

application assistant (Twitter), 277, 282–285
application environment, defined, 14
application life cycles (Mojo), 25–27
Application Manager service (Mojo), 90–91
application menu commands, handling (Twitter), 304–305
application menu (webOS), 87–88
Aptana Studio plug-in (Eclipse), 29
`assert(s)` methods (Mojo), 47
attendee object (Calendar service), 94
attributes

- appinfo.json file, 38–40
- framework_config.json, 40–41

Audio service (Mojo), 93–94

■ B

background color, 128–129
Balsamiq Mockups, 108
banners, defined (webOS), 18
`Base64.decode()` method, 280

- basic auth (authentication), 275
- BBS (Bulletin Board System), 5–7
- bind() method, 37
- bindAsEventHandler() method, 102
- Book Time (Time Tracker application)
 - Button, 330
 - scene, 372–373
- Browser service (Mojo), 94
- btnSaveTap() method, 370, 372
- bubbling phase (events), 58
- buckets, defined, 56
- Button widget (Mojo), 68
- buttonfloat style, 147

C

- Calendar alarms, 92
- Calendar service (Mojo), 94–95
- Camera service (Mojo), 103
- capture phase (events), 58
- card view (webOS), 16
- categories (Code Cabinet application)
 - Category List scene, 131
 - CategoryAddAssistant() function, 138–139
 - CategoryListAssistant() function, 132–137
 - creating new, 120
 - deleting, 122
 - retrieving, 121
 - table, 112
- cell phones, history of, 5, 8–9
- chain methodology (events), 308
- Char package (Mojo), 50
- CheckBox widget (Mojo), 68–69, 371–372
- checkConnectivity() method, 179–180, 285, 300
- checkFeeds() method (Twitter), 289
- cleanup() method (scene assistant), 299
- clearing search fields (business search application), 199
- clone method, 384
- closeAllStages() method, 52
- closeStage() method, 52
- closures, 290
- cloud computing, 310
- Code Cabinet application
 - Add Category scene, 137
 - appinfo.json file, 113
 - categories, creating new, 120
 - categories, deleting, 122

- categories, retrieving, 121
- Category List scene, 131
- CategoryAddAssistant() function, 138–139
- CategoryListAssistant() function, 132–137
- CodeCabinet.js file, 115–116
- configuring, 113–114
- creating basic skeleton, 110–111
- DAO, initializing, 119–120
- DAO.js file, 116–119
- data model for, 111–113
- exercises to improve, 163–164
- features to implement, 107–108
- framework_config.json file, 114
- global-scope code, 114
- handling errors, 125
- planning application, 108–110
- Preferences scene, 161
- PreferencesAssistant() function, 162–163
- Search Results scene, 155–156
- Search scene, 153–154
- SearchAssistant() function, 154–155
- SearchResultsAssistant() function, 156–160
- Snippet Details scene, 141–142
- Snippet List scene, 139–140
- SnippetDetailsAssistant() function, 145–153
- SnippetListAssistant() function, 140–141
- snippets, creating, 122
- snippets, deleting, 124–125
- snippets, retrieving, 122–124
- snippets, updating, 124
- sources.json file, 113–114
- stage, setting, 125–128
- stylesheet for (codecabinet.css), 128–129
- view HTML for Add Category scene, 137–138
- view HTML for Category List scene, 131–132
- view HTML for Preferences scene, 161–162
- view HTML for Search Results scene, 156
- view HTML for Search scene, 154
- view HTML for Snippet Details scene, 142–145
- view HTML for Snippet List scene, 140
- view HTML for Welcome scene, 129–130
- Welcome scene, 129
- WelcomeAssistant() function, 130–131

- Code Drawer, 143
- Collection class (Java), 320

commander chain, 126
 commands (business search application)
 Details scene, 214–217
 Search scene, 197
 configuring
 Code Cabinet application, 113–114
 Engineer game application, 231–233
 local business search application, 175–176
 Time Tracker application, 333
 Twitter monitor application, 279–280
 Connection Manager service (Mojo), 95, 180
 constants of GameScreenAssistant class, 243–245
 Contacts service (Mojo), 96–97
 Controller package (Mojo), 50–52
 convertToNode() method (Mojo.view), 65
 Cookie package (Mojo.Model), 63–64
 cookies (webOS), 20
 createCategory() method, 120, 139
 createSnippet() method, 122
 createStageWithCallback() method, 52
 critical chain approach (project management), 308
 critical path algorithm, 308
 cross fades, 65
 crossFade transitions, 349
 CSS (Cascading Style Sheets), 242
 CustomField object (Contacts service), 96

■ **D**

DAO, initializing
 Code Cabinet application, 119–120
 Time Tracker application, 335–336
 dao.deleteCategory() method, 135
 DAO.js file
 Code Cabinet application, 116–119
 Time Tracker application, 314
 dao.update() method, 376
 dark theme (Palm), 181
 dashboard (webOS), 18–20
 data fields, app assistant (Twitter), 282–283
 data, formatting, 63
 data model
 for Code Cabinet application, 111–113
 for Engineer game application, 231
 for local business search application, 174–175

 for Time Tracker application, 331
 for Twitter monitor application, 279
 Date class (Java), 321
 Date object
 JavaScript, 342
 java.util package, 317
 DatePicker widget (Mojo), 72, 362–363
 debounce, defined (Mojo), 60
 debugging, on Palm Pre, 40–41
 decorator object, defined, 62
 deleting
 delete keyword (JavaScript), 221
 deleteCategory() method, 122
 deleteProject() method, 359
 deleteSnippet() method, 124, 152
 entities (Time Tracker application), 341–342
 dependencies (project management), 387
 deploying GAE application, 313–314
 Depot, opening (business search application), 177–179
 Depot wrapper (Mojo), 55–57
 DetailsAssistant() function (business search application), 209–212
 developer mode in Palm Pre, 33
 development, webOS, 27–28
 device orientation change events (business search application), 188–190
 Document Viewers service (Mojo), 97
 document.getElementById() method, 53
 doDelete() method, 327
 doGet() method, 326
 doPost() method, 322
 downloading GAE, 311
 Drawer widget (Mojo), 84–85, 142–144
 drawerModels object, 145–146
 drawing (Engineer game application)
 control console and hands, 258–259
 frame and flashing lights, 257
 game-play area, 259
 DTO (data transfer object) example, 316–318

■ **E**

Eclipse, installing, 29
 Email service (Mojo), 101–102
 EmailAddress object (Contacts service), 96
 emulator, webOS, 28
 enableFullscreenMode() method, 53, 348

encrypt/decrypt methods (Mojo.model), 63
 engine, heating up (Engineer game application), 264–265
 Engineer game application
 appinfo.json file, 232–233
 configuring, 231–233
 creating basic skeleton, 229–231
 data model for, 231
 directory structure/file contents, 230
 exercises to improve, 271
 framework_config.json file, 233
 Game Over dialog box, 268
 GameOverAssistant(inAssistant) function, 269–271
 gameScreen scene, 241
 gameScreen scene, activating, 251–253
 GameScreenAssistant class, 243–248
 global-scope code, 234
 K&G Arcade and, 228
 main game loop. *See* main game loop (Engineer application)
 overview, 225
 planning application, 228–229
 player input, handling, 266–267
 playing sound, 256–257
 requirements for, 225–228
 sources.json file, 233
 stage, setting, 234
 starting new game, 253–256
 stylesheet for, 234–235
 titleScreen scene, 235
 TitleScreenAssistant() function, 238–241
 view HTML for Game Over dialog box, 268–269
 view HTML for gameScreen scene, 241–243
 view HTML for titleScreen scene, 235–238
 entities (Time Tracker application)
 creating, 336–338
 deleting, 341–342
 retrieving, 338
 updating, 341
 error handling (Code Cabinet application), 125
 events
 event bubbling (propagation), 58
 event object (Calendar service), 94
 Event package (Mojo), 57–59

Event.requiresProperties() method (Mojo), 66
 Event.visible() method (Mojo), 66
 shake events, 102–103
 executeSql() methods, 120
 exercises
 for Code Cabinet application, 163–164
 for Engineer game application, 271
 for local business search application, 225
 for Time Tracker application, 386–387
 for Twitter monitor application, 305

■ F

Favorites scene (business search application), 219–221
 feeds, monitoring (Twitter application), 289–291
 FilterField widget (Mojo), 76–77
 FilterList widget (Mojo), 77
 flicking, defined (WebOS), 16
 Format package (Mojo), 59–60
 formatDate() method, 366, 384
 framework_config.json file
 business search application, 176
 Code Cabinet application, 114
 configuration options in, 40–41
 Engineer game application, 233
 Twitter application, 280
 friends (Twitter application)
 getting list from Twitter, 299–300, 302–303
 getting list of, 275–276
 function, defined in JavaScript, 44
 Function package (Mojo), 60–61

■ G

GAE (Google App Engine)
 account sign up, 311
 creating simple application, 311–313
 deploying application, 313–315
 downloading and setup, 311
 overview, 309–311
 Game Over dialog box, 268
 GameOverAssistant(inAssistant) function, 269–271
 gameScreen scene, 241, 251–253
 GameScreenAssistant class, 243–248
 Gantt chart, 308

Garrett, Jesse James, 168
 gestures (webOS), 16–17
 GET method, 275
 getActiveStageController() method, 52
 getAppController() method, 53
 getRequestURI() method, 323
 getSceneController() method, 53
 getScenes() method, 53
 getScreenOrientation() method, 52
 getter and setter methods, 318
 global-scope code
 Code Cabinet application, 114
 Engineer game application, 234
 local business search application, 176–181
 Time Tracker application, 333
 Twitter monitor application, 280
 Google Web Toolkit, 312
 GPS fix, getting (business search application), 197–199
 GPS service (Mojo), 97–98
 graphical rollovers (JavaScript), 249
 greet() method, 36

■ **H**

handleLaunch() method (app assistant), 283–284
 handleOrientation() method, 189
 handling errors (Code Cabinet application), 125
 heating up engine (Engineer game application), 264–265
 HelloWorld application, building, 30–42
 hintText, 363
 HTC Mogul cell phone, 9
 HTML5 storage (webOS), 20
 HttpServlet class (Java), 321
 HttpServletRequest class (Java), 321
 HttpServletResponse class (Java), 321
 Hungarian Notation, 148

■ **I**

IDE (integrated development environment), Eclipse, 29
 images
 directory, 174–175
 image field member in
 GameScreenAssistant, 247–248
 ImageView widget (Mojo), 85–86

 used in Engineer game application, 231
 IMName object (Contacts service), 96
 index.html file, 23
 inEvent object, 126
 Info Drawer, 146
 info() method, 322
 init() method, 120, 335, 345
 installing Eclipse, 29
 IntegerPicker widget (Mojo), 73–74, 195
 Internet, history of, 5–7
 IOException class (Java), 320
 iPhones
 developing for, 10
 history of, 9–10
 isActiveAndHasScenes() method, 53
 isArray() method, 296
 Item Details scene (business search application), 208–209

■ **J**

Java
 Java 6 runtime, 28
 java.net.URLDecoder() class, 323
 JavaScript
 Date object, 342
 JavaScript.utils package, 44
 source code, 45–46
 JDO (Java data objects), 317, 321
 JsonSerializer class, 320

■ **K**

K&G Arcade game, 228
 Keywords Drawer, 144
 Konami code, 33

■ **L**

launch() method, 52
 Launcher (webOS), 17
 launching applications (app assistant), 283–288
 light theme (Palm), 181
 list of friends, getting (Twitter application), 275–276
 List templates (Twitter application), 298–299
 List widget (Mojo), 74–76
 listen() method, 53
 ListSelector widget (Mojo), 71–72
 load balancing (GAE), 310

loadStylesheet() method, 53
 local business search application
 activating/deactivating Details scene, 212–214
 checkConnectivity() method, 179–180
 clearing search fields, 199
 configuration of, 175–176
 creating basic skeleton, 173–175
 data model for, 174–175
 DetailsAssistant() function, 209–212
 device orientation change events, 188–190
 exercises to improve, 225
 Favorites scene, 219–221
 features of, 167
 global-scope code, 176–181
 GPS fix, getting, 197–199
 handling commands (Details scene), 214–217
 handling commands (Search scene), 197
 Item Details scene, 208–209
 LocalBusinessSearch.js, 176–178
 map image, getting (Details scene), 217–219
 mashups, 165–167
 opening Depot, 177–179
 orientation changes (Details scene), 219
 planning application, 172–173
 Preferences scene, 221–222
 PreferencesAssistant() function, 223–225
 returned search results, handling, 206–208
 Search Results scene, 201–202
 Search scene, 190–191
 SearchAssistant() function, 193–197
 searches, performing, 200–201
 SearchResultsAssistant() function, 203–206
 stage, setting, 181–183
 stylesheet for, 183–185
 user selection of search results, handling, 208
 view HTML for Item Details scene, 209
 view HTML for Preferences scene, 222–223
 view HTML for Search Results scene, 202
 view HTML for Search scene, 191–193
 view HTML for Welcome scene, 186–187
 Welcome scene, 185–186
 WelcomeAssistant() function, 187–188
 local data storage (webOS), 20
 LocalBusinessSearch.js, 176–178

Log In Dialog scene (Time Tracker application), 352
 Log package (Mojo), 61–62
 Logger class (Java), 321
 LogInAssistant(inAssistant) function (Time Tracker application), 353–356

■ M

main game loop (Engineer application)
 drawing control console and hands, 258–259
 drawing frame and flashing lights, 257
 drawing game-play area, 259
 heating up engine, 264–265
 main game logic, 259–263
 particle direction, changing, 263–264
 updating player scores, 265–266
 Main scene (Twitter application)
 activating, 288–289
 overview, 284–285
 setup, 286–287
 MainAssistant() function (Twitter application), 286
 mainLoop() method, 263
 map image, getting (business search application), 218–219
 Map service (Mojo), 98–99
 mashups, 165–167
 menu commands, handling (Twitter application), 304–305
 menus, webOS, 87–89
 Messaging Service, Mojo, 99, 104
 method, defined in JavaScript, 44
 micro-blogging sites, 274
 Model package (Mojo), 62–63
 Model-View-Controller (MVC) pattern, 22–23
 modelChanged() event, 53
 Model.Cookie package (Mojo), 63–64
 Mojo framework
 API overview, 43–45
 application life cycles, 25–27
 Mojo Depot data storage (webOS), 20
 Mojo.Animation, 46–47
 Mojo.assert(s) methods, 47
 Mojo.Char package, 50
 Mojo.Controller package, 50–52

- Mojo.Controller.AppController methods, 52–53
- Mojo.Controller.SceneController, 53–55
- Mojo.Controller.StageController, 53
- Mojo.Depot, 55–57
- Mojo.Event package, 57–59
- Mojo.Format package, 59–60
- Mojo.Function package, 60–61
- Mojo.Log package, 61–62
- Mojo.Model package, 62–63
- Mojo.Model.Cookie package, 63–64
- Mojo.require(s) methods, 47–51
- Mojo.Service package, 64
- Mojo.Transition package, 64–65
- Mojo.View package, 65–66
- Mojo.Widget package, 66
- overview, 21–22
- scene stacks, 25
- scenes (views and assistants), 24–25, 35
- stages, 24
- standard directory structure, 23–24
- Mojo Messaging Service, 103–104
- Mojo Namespace, 47–49
- Mojo scene wizard, 111
- Mojo.Controller.errorDialog() method, 139
- Mojo.Event.keypress event, 251–252
- Mojo.Event.listen() method, 36, 188–189
- Mojo.Event.stageActivate event, 238–240
- Mojo.Function.debounce() method, 60–61
- Mojo.Function.Synchronize class, 61
- Mojo.View.render() method, 384–385
- monitoring feeds (Twitter application), 289–291
- Monotonic alarms, 92
- multitasking feature (webOS), 15

N

- Namespace, Mojo, 47–49
- navigation and gestures (webOS), 16–17
- Notes Drawer, 143
- notification bar (webOS), 18

O

- OkResponse/ErrorResponse DTOs, defining, 318–319
- orientation changes, handling (business search application), 219
- ORM (object-relational mapping), 321

- out-of-band requests, 167

P

- packages, defined (JavaScript), 43
- Palm
 - Developer Network (PND), 28, 132
 - Inspector tool, 254–255
 - palm-page-header style, 162
 - PalmPilot PDAs, 7–8
- Palm Pre
 - debugging on, 40–41
 - Engineer game application for, see Engineer game application
 - overview, 11–12
- ParseException class (Java), 320
- particles (Engineer game application)
 - changing direction of, 264
 - movement of, 260–262
 - states of, 260
 - types of, 227
 - use in game, 236
- PasswordField widget (Mojo), 79–80
- PDAs, history of, 3, 7
- PersistenceManager class (Java), 321
- Preferences scene
 - business search application, 221–222
 - Code Cabinet application, 110, 161
- PreferencesAssistant() function
 - business search application, 223–225
 - Code Cabinet application, 162–163
- projects (Time Tracker application)
 - creating/updating, 322–325
 - deleting, 327
 - Project Summary scene, 382
 - projectSelected() method, 375
 - retrieving, 326–327
- Prototype JavaScript library, 37, 322
- pseudo-constants (JavaScript), 45
- pushScene() method, 53, 295
- PUT method, 316
- PuTTY client, 37

Q

- Query class (Java), 321
- Quick Launch bar (WebOS), 17

R

- RadioButton widget (Mojo), 70–71, 193
- removeAllBanners() method, 52–53
- render() method (Mojo.view), 65–66
- require(s) methods (Mojo), 47–51
- Resource Administration (Time Tracker application)
 - Button, 330
 - scene, 370
- resources (Time Tracker application)
 - defined for project management, 315–316
 - Resource Details scene (Time Tracker application), 370–372
 - Resource leveling (project management), 308
 - Resource Summary scene, 385–386
- REST (Representational State Transfer), 274–275, 315–317
- retrieveCategories() method, 121
- retrieveSnippets() method, 122
- retrieving entities (Time Tracker application), 338–341
- returned search results, handling (business search application), 206–208
- RichTextEdit widget (Mojo), 80

S

- Safari web browser, 10
- same-domain policy (AJAX requests), 168–169
- saving changes (Twitter application), 303
- scaling, GAE and, 310
- scene assistant (Time Tracker application)
 - for Book Time scene, 373–377
 - for Project Administration scene, 357–359
 - for Project Details scene, 361–367
 - for Project Summary scene, 383–385
 - for Summaries scene, 378–382
 - for Task Details scene, 368–370
 - for Title scene, 348–349
 - for Welcome scene, 350–351
- scenes
 - scene controller (Mojo), 53–55
 - scene menus, 194
 - scene stacks (Mojo), 25
 - views and assistants (Mojo), 24–25, 35
- scores, updating (Engineer game application), 265–266

- scrim element, 191
- scrolling in WebOS, 17
- scrollInstructions() method, 240
- Scrum project methodology, 307
- SDK (software development kit)
 - GAE, 311
 - webOS, 28
- Search Results scene
 - business search application, 201–202
 - Code Cabinet application, 155–156
- Search scene
 - business search application, 190–191
 - Code Cabinet application, 153–154
- SearchAssistant() function
 - Code Cabinet application, 154–155
 - local business search application, 193–197
- searching (business search application)
 - performing searches, 200–201
 - search fields, clearing, 199
 - search results, handling, 206–208
 - search results, handling user selection of, 208
- SearchResultsAssistant() function
 - Code Cabinet application, 156–160
 - local business search application, 203–206
- send() method, 151
- serialize() method, 323
- server-side code (Time Tracker application), 316
- Service package (Mojo), 64
- serviceRequest() method, 54, 180
- services, Mojo framework
 - Accelerometer service, 102–103
 - Accounts service, 92
 - Alarms service, 92–93
 - Application Manager service, 90–91
 - Audio service, 93–94
 - basics, 22, 89–90
 - Browser service, 94
 - Calendar service, 94–95
 - Camera service, 103
 - Connection Manager service, 95
 - Contacts service, 96–97
 - Document Viewers service, 97
 - Email service, 101–102
 - GPS service, 97–98
 - Map service, 98–99

- Messaging service, 99
- Mojo Messaging Service, 103–104
- Phone service, 99
- Photos service, 99–100
- System Properties service, 100
- System Service service, 100
- System Sounds service, 101
- Video service, 101
- View File service, 101
- servlet for HTTP requests (Time Tracker application), 320–321
- setDefaultTransition() method, 54
- setInitialFocusedElement() method, 54
- setInterval()/setTimeout() methods, 288
- Settings scene assistant (Twitter application), 294–298
- Settings scene (Twitter application), 292
- setup() method (Mojo), 27, 147, 162–163
- setupWidget() method, 36, 54, 363
- setUserIdleTimeout() method, 54
- setWidgetModel() method, 54
- shake events, 102–103
- showAlertDialog() method, 127, 152
- showBanner() method, 52–53
- simpleAdd() method, 303
- SimpleDateFormat class (Java), 320–324
- Slider widget (Mojo), 83–84, 192–193
- smartphones, history of, 8
- snippets (Code Cabinet application)
 - Snippet Details scene, 110, 141–142
 - Snippet List scene, 139–140
 - snippetDescriptor object, 150
 - SnippetDetailsAssistant() function, 145–153
 - SnippetListAssistant() function, 140–141
 - table, 112
- snippets of code
 - creating, 122
 - deleting, 124–125
 - retrieving, 122–124
 - updating, 124
- SOA (service-oriented architecture), 166
- sounds
 - playing (Engineer game application), 256–257
 - system, 101
 - System Sounds service (Engineer game application), 227
- source code, JavaScript, 45–46
- sources.json file, 23
 - Code Cabinet application, 113–114
 - Engineer game application, 233
 - Twitter application, 279–280
- Spinner widget (Mojo), 82–83
- split() method (strings), 291
- SQL statements
 - in Code Cabinet application, 87
 - for projects/tasks/resources, 102
- SQLite Manager (Firefox extension), 112
- SSH connections, 37
- stages (Mojo)
 - basics of, 24
 - stage controller, 27, 53
 - StageAssistant class, 126
- stages, setting
 - Code Cabinet application, 125–128
 - Engineer game application, 234
 - local business search application, 181–183
 - Time Tracker application, 33565
 - Twitter monitor application, 281
- standard directory structure (Mojo), 23–24
- startGame() method, 251–252, 256, 271
- static, defined (Java), 320
- status bar (webOS), 17–18
- strip() method, 139
- style classes
 - Mojo, 132
 - Twitter monitor application, 281–282
- stylesheets
 - for Code Cabinet application, 128–129
 - for Engineer game application, 234–235
 - for local business search application, 183–185
 - for Time Tracker application, 345–346
- Summaries Button, 330
- Summaries scene (Time Tracker application), 376–377
- swapScene() method (stage controller), 295, 303
- swiping, defined (WebOS), 16
- synchronize() method, 343
- System Properties service (Mojo), 100
- System Service service (Mojo), 100
- System Sounds service, 101, 227, 256

■ **T**

- tapping, defined (WebOS), 16
- Task Details scene (Time Tracker application), 367–368
- Task Summary scene (Time Tracker application), 385
- TaskServlet/ResourceServlet (Time Tracker application), 328
- templates, List (Twitter application), 298–299
- TextField widget (Mojo), 77–79
- this.controller.setupWidget(), 36
- Time Tracker application
 - Book Time scene, 372–373
 - configuring, 333
 - creating basic skeleton, 331
 - DAO, initializing, 335–336
 - DAO.js file, 333–335
 - data model for, 331–333
 - directory structure/file contents, 331
 - DTO example, 316–318
 - entities, creating, 336–338
 - entities, deleting, 341–342
 - entities, retrieving, 338–341
 - entities, updating, 341
 - exercises to improve, 386–387
 - features of, 307–309
 - global-scope code, 333
 - Google App Engine (GAE). *See* GAE (Google App Engine)
 - Log In Dialog scene, 351
 - LogInAssistant(inAssistant) function, 353–356
 - OkResponse/ErrorResponse, defining, 318–319
 - planning application, 329–330
 - Project Administration scene, 356–357
 - Project Details scene, 359–360
 - Project Summary scene, 381–382
 - projects, creating/updating, 322–325
 - projects, deleting, 327
 - projects, retrieving, 326–327
 - Resource Administration scene, 370
 - Resource Details scene, 370–372
 - Resource Summary scene, 385–386
 - REST and. *See* REST (Representational State Transfer)
 - scene assistant for Book Time scene, 373–376
 - scene assistant for Project Administration scene, 357–359
 - scene assistant for Project Details scene, 361–367
 - scene assistant for Project Summary scene, 383–385
 - scene assistant for Summaries scene, 378–381
 - scene assistant for Task Details scene, 368–370
 - scene assistant for Title scene, 348–349
 - scene assistant for Welcome scene, 350–351
 - server-side code, 316
 - servlet for HTTP requests, 320–321
 - SQL statements for
 - projects/tasks/resources, 333–335
 - stage, setting, 345
 - stylesheet for, 345–346
 - Summaries scene, 377
 - Task Details scene, 367–368
 - Task Summary scene, 385
 - TaskServlet and ResourceServlet, 328
 - TimeTracker.js file, 342–345
 - Title scene, 347
 - utilities class, 319–320
 - view HTML for Book Time scene, 373
 - view HTML for Log In Dialog scene, 352–353
 - view HTML for Project Administration scene, 357
 - view HTML for Project Details scene, 360–361
 - view HTML for Project Summary scene, 382–83
 - view HTML for Summaries scene, 377–378
 - view HTML for Task Details scene, 368
 - view HTML for Title scene, 347–348
 - view HTML for Welcome scene, 350
 - web.xml configuration file, 328–329
 - Welcome scene, 349–350
- TimePicker widget (Mojo), 73
- Title scene (Time Tracker application), 347
- titleScreen scene (Engineer game application), 235
- TitleScreenAssistant() function (Engineer game application), 238–241

ToggleButton widget (Mojo), 69–70
 toString() method, 318
 transaction() method, 120
 Transition package (Mojo), 64–65
 Twitter monitor application
 application assistant, 282–285
 application menu commands, handling, 304–305
 cleanup() method (scene assistant), 299
 configuring, 279–280
 creating basic skeleton, 278
 data model for, 279
 exercises to improve, 305
 features of, 273–274
 friends list, getting from Twitter, 299–300, 302–303
 global-scope code, 280
 List templates, 298–299
 Main scene, 284–285
 Main scene, activating, 288–289
 Main scene setup, 286–287
 MainAssistant() function, 286
 monitoring feeds, 289–291
 overview, 273
 planning application, 276–278
 REST and, 274–275
 saving changes, 303
 Settings scene, 292
 Settings scene assistant, 294–298
 stage, setting, 281
 style classes for, 281–282
 Twitter web services, 274
 user verification, 275
 users, getting current status of, 276
 verifying Twitter account, 300–301
 view HTML for Main scene, 285
 view HTML for Settings scene, 292–293
 txtCategoryNameModel, 138

U

UI design principles (webOS), 20–21
 UI System Manager (webOS), 21–22
 updating
 entities (Time Tracker application), 341
 player scores (Engineer game application), 265–266
 updateSnippet() method, 124

useLandscapePageUpDown() method, 54
 users
 getting current status of (Twitter application), 276
 interface (webOS), 14–15
 user selection of search results (business search application), 208
 verification (Twitter application), 275
 utilities class (Time Tracker application), 319–320

V

verifying Twitter account, 300–301
 Video service (Mojo), 101
 View File service (Mojo), 101
 view HTML (Code Cabinet application)
 for Add Category scene, 137–138
 for Category List scene, 131–132
 for Preferences scene, 161–162
 for Search Results scene, 156
 for Search scene, 154
 for Snippet Details scene, 142–145
 for Snippet List scene, 140
 for Welcome scene, 129–130
 view HTML (Engineer game application)
 Game Over dialog box, 268–269
 gameScreen scene, 241–243
 titleScreen scene, 235–238
 view HTML for business search application
 Item Details scene, 209
 Preferences scene, 222–223
 Search Results scene, 202
 Search scene, 191–193
 Welcome scene, 186–187
 view HTML (Time Tracker application)
 for Book Time scene, 373
 for Log In Dialog scene, 352–353
 for Project Administration scene, 357
 for Project Details scene, 360–361
 for Project Summary scene, 382–283
 for Summaries scene, 377–378
 for Task Details scene, 368
 Title scene, 347–348
 Welcome scene, 350
 view HTML (Twitter application)
 Main scene, 285
 Settings scene, 292–293

View package (Mojo), 65–66
VirtualBox, 28

■ **W**

watchModel() method, 54–55
Web Services, 166
web services, Twitter, 274
web sites, for downloading
 Aptana Studio plug-in (Eclipse), 29
 Balsamiq Mockups, 108
 Base64.js file code, 280
 Eclipse, 29
 GAE SDK, 311
 Java 6 runtime, 28
 K&G Arcade game, 228
 SQLite Manager (Firefox extension), 112
 VirtualBox, 28
 webOS SDK, 28
web sites, for further information
 Adaptive Path, 168
 Yahoo! Maps APIs, 172
WebKit, 21–22
webOS (Palm)
 activities feature, 15–16
 application, creating in Eclipse, 30–42
 card view, 16
 dashboard, 18–20
 development, 27–28
 Engineer game application for. *See* Engineer
 game application
 Launcher, 17
 local data storage, 20
 menus, 87
 multitasking feature, 15
 navigation and gestures, 16–17
 overview, 12–14
 pop-ups, 18
 SDK, 28
 status bar, 17–18
 structure of, 14
 UI design principles, 20–21
 user interface, 14–15
WebView widget (Mojo), 86–87
web.xml configuration file (Time Tracker
 application), 328–329
Welcome scene
 business search application, 185–186

Code Cabinet application, 110, 129
Time Tracker application, 349–350
WelcomeAssistant() function
 Code Cabinet application, 130–131
 local business search application, 187–188
widgets (Mojo framework)
 basics, 66–67
 Button widget, 68
 CheckBox widget, 68–69
 DatePicker widget, 72
 defined, 22
 Drawer widget, 84–85
 FilterField widget, 76–77
 FilterList widget, 77
 ImageView widget, 85–86
 IntegerPicker widget, 73–74
 List widget, 74–76
 ListSelector widget, 71–72
 menus, 87–89
 PasswordField widget, 79–80
 ProgressBar widget, 80–81
 ProgressPill widget, 81–82
 RadioButton widget, 70–71
 RichTextEdit widget, 80
 Slider widget, 83–84
 Spinner widget, 82–83
 TextField widget, 77–79
 this.controller.setupWidget(), 36
 TimePicker widget, 73
 ToggleButton widget, 69–70
 WebView widget, 86–87
 Widget package, 66
Windows CE, defined, 8
Wireless Access Protocol (WAP) browsers, 8

■ **X**

x-mojo-element attribute, 67

■ **Y**

Yahoo!
 Local Search service, 169–170
 Maps image service, 171–172
 Web Services, 169

■ **Z**

zip code fields, 196
zoom fades, 64–65
zoomFade transition, 188