

# Index

## NUMBERS

**411Sync, 39–40**

## A

**A9.com, registering with, 279–280**

`abort ( )` method, **168**

**aborting browsers requests, 168**

**access keys, ECS FAQ, 340–342**

`addListenerEvent ( )` method, **168**

**AdSense API, 36**

**advertisement platform, Google, 36**

**aggregating blogs, 33–34**

**Ajax. See also wish lists**

definition, 179

libraries, 206–207

mashing with

architecture, 90

functional description, 90–91

pros and cons, 91–92

when to use, 92

pros and cons, 91–92, 182

**Alexa, 26–28**

`AlternatingItemTemplate`, **132**

**Amazon**

See also browsers

See also JSON Amazon customer browser

See also mashups, creating

See also wish lists

associates, 22, 343–344

Begin tasks, 274

categorizing data, 174–175. See also

`BrowseNodes`

consumer sites, 10

data, retrieving, 103–105

developers program, 10–11. See also Developer

Web; developers program

End tasks, 274

error handling, 158

finding restaurants, 221–222. See also restaurant

finder

online store, 333–334

product database, accessing. See ECS

(E-Commerce Service)

REST URI, validating, 163–164

search interface

accurate XPath, 143

converting the Yahoo query, 141–142

default namespace, 143

namespaces, 143

performance, 144

real time results, 330–332

templates, triggering, 143

XSL output type, specifying, 143

XSL style sheet, applying, 147–149

XSL style sheet, description, 144–147

XSL transformations, 142–144

Zuggest, 330–332

Turing Test, 334

and Web 2.0, 9–11

Web services. See Developer Web site; ECS

(E-Commerce Service)

## Amazon mobile

- architecture, 284
- Blackberry emulator, installing, 285–286
- functional description, 284–285
- MIME types, configuring, 290–294
- search page, creating, 290–294
- WAP (Wireless Access Protocol), 285
- WML (Wireless Markup Language)
  - detail page, creating, 297–299
  - forms, 287–289
  - home page, 290–294
  - list page, creating, 294–297
  - pages, 287–289
  - transforming Amazon data into, 289–290
- `AmazonUtility` class, adding, 109–113
- analysis, Web sites, 26–28
- ancestor nodes, 19
- application development. *See* development program
- application settings, 102–103
- artificial artificial intelligence, 24
- ASIN (Amazon Standard Identification Number), 17–18
- associates of Amazon, 22
- `Async` client, creating, 262–267
- asynchronous communication, browsers, 158, 172
- asynchronous techniques, 258–260
- Atlas, 335
- Atom, 79
- Audio Search, Yahoo, 30
- authentication, S3, 308
- author, RSS element, 115
- auto-completion, wish lists, 206
- AWS (Amazon Web Service), 105–108
- `AWSAccessKeyId` parameter, 307–310

## B

- Baby Registries. *See* wish lists
- BaeBo, 333–334
- BBC services, 40
- Begin task
  - Amazon, 274
  - Google, 272–274
- binding data to controls, 132–133
- Blackberrys. *See* Amazon mobile
- Blogger API, Google, 36
- blog-rolls, 33–34

## blogs

- aggregating, 33–34
- hosting, Google, 36
- monitoring, 33–34
- bookmarks. *See* links
- borders, maps, 217–219
- `BrowseNodeID` property, 19
- `BrowseNodeLookup` operation, 17

## BrowseNodes

- ancestor nodes, 19
- `BrowseNodeID` property, 19
- child nodes, 19
- definition, 18
- Name property, 19
- properties, 19
- searching for, 17
- uses for, 19

## browser caching, breaking, 204

## browser requests

- aborting, 168
- creating, 169
- event handlers, 168–169
- HTTP headers, 168–169
- overriding `MimeType`, 169
- sending, 169–170
- triggering events, 168
- `XmlHttpRequest` object, 168–169

## browsers

- Amazon errors, 158
- Amazon REST URI, validating, 163–164
- architecture, 156–157
- asynchronous communication, 158, 172
- categorizing data, 174–175. *See also* `BrowseNodes`
- data providers, 160–162
- ECS FAQ, 345–347
- functional description, 157
- `getdata.aspx`, 161–162, 172
- JAH (Just Asynchronous HTML), 158
- named templates, 175–176
- namespace issues, 176
- network proxies, 164
- parsing a tree structure, 175–176
- placeholder, 159
- port assignment, 166
- `readyState` property, 171–172
- recursion, 175–176
- recursive XSL templates, 175–176

- restaurant finder, 219–220
- rich browser experience, 7
- security issues, 163–164
- Web server response, 171–172
- wish list support, 206
- XSL tools
  - if-then-else statements, 165
  - templates, matching to multiple elements, 165
  - XmlHttpRequest object, 168–169
  - YouTube response, transforming, 165–168

### **browsers, Amazon customers**

- architecture, 236
- Dynamic Script method, 238–240
- functional description, 236–238
- Hello World example, 240–241
- notation, 242
- public APIs
  - advantages of, 243
  - versus Ajax, 243
  - creating the browser, 244–245
  - getting from Amazon, 243–245
  - processing the JSON object, 250–252
  - searching for customers, 245–247
  - testing, 252–253
  - XSL file, 247–250

### **buckets**

- creating, 310–311
- definition, 306
- deleting, 312
- listing, 311–312
- listing contents of, 315–316

### **BuildAmazonURI**

- modifying, 187–188
- restaurant finder example, 229

### **business opportunities, 22–23**

## **C**

### **cards, 81–82**

- CartAdd **operation, 16, 198, 201**
- CartClear **operation, 16, 198, 201**
- CartCreate **operation, 16, 198, 199–200**
- CartGet **operation, 16, 198, 200–201**
- CartId, **storing, 201–203**
- CartModify **operation, 16, 198**
- CartNewReleases, ResponseGroup, **21**
- CartSimilarities, ResponseGroup, **21**

### **Cascading Style Sheets (CSS). See CSS (Cascading Style Sheets)**

#### **case sensitivity, CSS, 62**

**categories of products. See BrowseNodes**

**category, RSS element, 115**

#### **cell phones**

- Internet access, 81. *See also* WML (Wireless Markup Language)
- searching from, 39–40

#### **channels, RSS, 113**

#### **child nodes, 19**

#### **classifying links, 32–33**

#### **collages, 325–329**

**comments, RSS element, 115**

#### **comparison shopping, 25. See also Amazon mobile**

#### **computing power, scalable, 25**

#### **connectionless HTTP, 68**

#### **consumer sites, Amazon, 10**

#### **consuming RSS data, 120–121**

#### **Content Analysis, Yahoo, 31**

#### **controls, binding to data, 132–133**

#### **convergence, 6**

#### **cookies, 202–203**

CreateBucket **operation, 310–311**

#### **creating**

- Amazon mobile search page, 290–294
- Async client, 262–267
- browser requests, 169
- browsers requests, 169
- buckets, 310–311
- forms, 99–101
- list pages, 294–297
- overlays, 229
- remote carts, 198, 199–200
- RSS generator, 109–113, 270–271
- search page, 290–294
- test Web service, 260–267
- WML detail page, 297–299
- WML list page, 294–297
- XML documents, 115–120

#### **creating, a remix**

- AmazonUtility class, adding, 109–113
- application settings, 102–103
- architecture, 97–98
- configuration data, storing, 102–103
- FormatAsHTML function, 107–108

### creating, a remix (continued)

#### forms

- creating, 99–101
- modifying, 111–112

#### functional description, 98

invoking AWS (Amazon Web Service), 105–108

`ItemSearch` request, 105–108

retrieving Amazon data, 103–105

#### RSS 2.0

- adding a feed, 121–122
- channels, 113
- consuming RSS data, 120–121
- documents, sample, 113–114
- `FormatAsRSS` function, 117–118
- items, 113
- mandatory fields, 115
- optional fields, 115
- outputting RSS, 117–120
- search output, sample, 119–120
- switching locales, 122–124
- validating RSS, 122
- XML documents, creating, 115–120
- XML elements, 114

RSS generator, creating, 109–113

search entry screen, code, 100–101

#### search results

- converting to HTML, 107–108
- iterating through, 107–108

`SearchAmazon` function, 105–106, 105–108

searching Amazon, 105–108

server-side code, 102–103

SOAP proxy class, 103–104, 105–108

switching locales, 104–105

UDDI (Universal Description, Discovery, and Integration), 104

user interface, 98–102

`web.config` file, 102–103

### creating, mashups

#### Amazon search interface

- accurate XPath, 143
- converting the Yahoo query, 141–142
- default namespace, 143
- namespaces, 143
- performance, 144
- templates, triggering, 143
- XSL output type, specifying, 143
- XSL style sheet, applying, 147–149

XSL style sheet, description, 144–147

XSL transformations, 142–144

approach to, 131

architecture, 129–130

binding data to controls, 132–133

functional description, 130

mashing search results, 149–152

`Repeater` control, 131–133

#### user interface

- footers, defining, 132–133
- formatting alternate rows, 132–133
- headings, defining, 132–133
- item separators, 132–133
- search results, displaying, 131–133
- templates, 132–133

#### Yahoo search interface

- generic helper class, 137–139
- HTML, retrieving, 134–137
- HTTP `GET` request, 134
- mini-browser, C#, 135–137
- results, sample, 134
- serialization, 139–141
- XML, retrieving, 137–139
- XML documents, converting to a `DataSet`, 139–141
- Yahoo REST URI example, 134–137

### CSS (Cascading Style Sheets). See also Master

#### Pages; skins; themes

- case sensitivity, 62
- issues, 63–64
- style sheets, adding, 62–63
- `z-index` attribute, 217–219

### customer content, searching for, 16

#### Customer Content data type, 14

#### customer content operations, 16

#### customer transactions, searching for, 17

`CustomerContentLookup` operation, 16

`CustomerContentSearch` operation, 16

`CustomerFull`, `ResponseGroup`, 21

## D

### data

- binding to controls, 132–133
- describing, 80–81
- providers, 160–162, 186
- representation, 83–84
- sources, 58
- syndication, 7, 77–78. See also Atom; RSS (Really Simple Syndication)

- “data is king,”** 6
  - data mining,** 23
  - data types, ECS,** 14
  - database of Amazon products. See ECS (E-Commerce Service)**
  - debugging**
    - restaurant finder example, 216
    - wish lists, 205
  - decks, 81–82**
  - delegates, 258–260**
  - `DeleteBucket` **operation,** 312
  - deleting**
    - S3 buckets, 312
    - S3 objects, 316–318
  - `del.icio.us` **site,** 32–33
  - delivery channels, multiple,** 7
  - description, RSS element,** 115
  - design cycles, 6–7**
  - developer ID. See `AWSAccessKeyId`**
  - Developer Web site. See also ECS (E-Commerce Service)**
    - Alexa, 26–28
    - artificial intelligence, 24
    - EC2 (Elastic Compute Cloud), 25
    - file storage, 25–26
    - Historical Pricing, 25
    - human intervention, 24. *See also* Mechanical Turk; Turing Test
    - main page, 23
    - Mechanical Turk, 24
    - queue service, 24–25
    - S3 (Simple Storage Service), 25–26
    - scalable computing power, 25
    - traffic analysis, 26–28
    - Web site analysis, 26–28
  - developers, individual,** 9
  - developers program, 13–14. See also ECS (E-Commerce Service)**
  - development environment, 53–54**
  - development program, 6–7**
  - development tools. See Microsoft Express Editions**
  - digests, 309–310**
  - `dispatchEvent ( )` **method,** 168
  - documentation, ECS,** 17
  - documents**
    - outline capture, 78–79
    - RSS, sample, 113–114
    - table of contents, 78–79
    - XML, creating, 115–120
  - documents, XML**
    - converting to a `DataSet`, 139–141
    - transforming
      - Amazon search interface, 142–144
      - description, 71–73
      - GeoRSS overlays, 225–227
      - restaurant finder example, 225–227
      - YouTube responses, 165–168
  - `DoGetCachedPage` **API,** 267–269
  - `DoGoogleSearch` **API,** 267–269
  - `DoSpellingSuggestion` **API,** 267–269
  - DTD (Document Type Definition), 69**
  - Dynamic Script method, 238–240**
- ## E
- eBay. See also wish lists**
    - API, 189
    - description, 38–39
    - REST interface, 190
  - EC2 (Elastic Compute Cloud), 25, 319**
  - ECS (E-Commerce Service). See also Developer Web site; developers program**
    - Amazon associates, 22
    - ASIN (Amazon Standard Identification Number), 17–18
    - `BrowseNodeLookup` operation, 17
    - `BrowseNodes`, 18–19
    - business opportunities, 22–23
    - `CartAdd` operation, 16
    - `CartClear` operation, 16
    - `CartCreate` operation, 16
    - `CartGet` operation, 16
    - `CartModify` operation, 16
    - `CartNewReleases`, `ResponseGroup`, 21
    - `CartSimilarities`, `ResponseGroup`, 21
    - core operations, 15–17
    - Customer Content data type, 14
    - customer content operations, 16
    - `CustomerContentLookup` operation, 16
    - `CustomerContentSearch` operation, 16
    - `CustomerFull`, `ResponseGroup`, 21
    - data types, 14
    - developers Web site, 14
    - documentation, 17
    - `EditorialReview`, `ResponseGroup`, 21

### **ECS (E-Commerce Service) (continued)**

- FAQs (frequently asked questions)
  - access keys, 340–342
  - accessing the ECS, 339
  - Amazon associates, 343–344
  - browsing, 345–347
  - detailed usage, 340–342
  - general questions, 340
  - identifiers, 340–342
  - remote cart, 344–345
  - resources, 347–348
  - searching, 345–347
- Help operation, 17
- Images, ResponseGroup, 21
- interface types, 14–15
- ItemLookup operation, 16
- ItemSearch operation
  - definition, 16
  - XML response, example, 20–21
- list operations, 16
- ListLookup operation, 16
- ListMinimum, ResponseGroup, 21
- ListSearch operation, 16
- locales, 15
- New Releases, ResponseGroup, 21
- Operation parameter, 19–21
- Product Data data type, 14
- product ID number, 17–18. *See also* ASIN (Amazon Standard Identification Number)
- product operations, 16
- query response, scope, 21–22
- remote cart operations, 16
- Request, ResponseGroup, 21
- required parameters, 19–21
- ResponseGroups, 21–22
- REST support, 14–15
- scenarios, 22–23
- searching for
  - BrowseNodes, 17
  - customer content, 16
  - customer transactions, 17
  - lists, 16
  - products, 16
  - third-party merchants, 17
- Seller Information data type, 14
- SellerListingLookup operation, 17
- SellerListingSearch operation, 17

- SellerLookup operation, 17
- Service parameter, 19–21
- Shopping Cart Contents data type, 14
- SimilarityLookup operation, 16
- SOAP support, 14–15
- Third-Party Product Information data type, 14
- TopSellers, ResponseGroup, 21
- TransactionLookup operation, 17
- EditorialReview, ResponseGroup, **21**
- Elastic Compute Cloud (EC2), 25, 319**
- elements, XML documents, 68–69**
- emptying remote carts, 198, 201**
- enclosure, **RSS element, 115**
- End **task**
  - Amazon, 274
  - Google, 272–274
- event handlers, browsers requests, 168–169**
- events**
  - triggering, 168
  - tying to map markers, 215–216
- Express Editions**
  - database development, 50
  - SQL Server 2005 Express Edition, 50
  - URL for, 49
  - Visual Basic 2005 Express Edition, 50–51
  - Visual C++ 2005 Express Edition, 50–51
  - Visual C# 2005 Express Edition, 50–51
  - Visual J# 2005 Express Edition, 51
  - Visual Web Developer 2005 Express edition, 50
  - Web development, 50
  - Windows development, 50–51
- Extensible Hypertext Markup Language (XHTML), 69–70**
- eXtensible Markup Language (XML). *See* XML (eXtensible Markup Language)**
- eXtensible Stylesheet Language (XSL), 71–73**

## **F**

### **FAQs (frequently asked questions), for ECS**

- access keys, 340–342
- accessing the ECS, 339
- Amazon associates, 343–344
- browsing, 345–347
- detailed usage, 340–342
- general questions, 340
- identifiers, 340–342
- remote cart, 344–345

resources, 347–348  
 searching, 345–347

#### feeds, RSS

adding, 121–122  
 converting to podcasts, 332

#### file storage. *See also* S3 (Simple Storage Service)

architecture, 304  
 authentication, 308  
 AWSSecretAccessKey parameter, 307–310  
 buckets  
   creating, 310–311  
   definition, 306  
   deleting, 312  
   listing, 311–312  
   listing contents of, 315–316  
 digests, 309–310  
 EC2 (Elastic Computing Cloud), 319  
 functional description, 304–305  
 HMAC (hash-based message authentication code), 310  
 HMACSHA1 algorithm, 309–310  
 message digests, 309–310  
 objects  
   definition, 306–307  
   deleting, 316–318  
   uploading, 313–314  
 permissions, 314–315  
 registering for, 306  
 required parameters, 307–310  
 secret keys, 307–308  
 signatures, 307–308, 309  
 timestamp, 308–309  
 usage, tracking, 318–319

#### FileUpload control, 313–314

#### Flickr, 34–35

#### FOAF (Friend of a Friend), 81

#### folksonomies, 4–5

#### FooterTemplate, 132

#### FormatAsHTML function, 107–108

#### FormatAsRSS function, 117–118

#### formatting, overriding

*See* CSS (Cascading Style Sheets)  
   *See* Master Pages  
   *See* skins  
   *See* themes

#### forms

creating, 99–101  
 modifying, 111–112  
 WML (Wireless Markup Language), 287–289

#### 411Sync, 39–40

function pointers. *See* delegates

## G

#### gadget manifest, 335–336

#### gadgets, 335–336

#### Gadgets API, Google, 36

#### generic helper class, 137–139

#### geo-coding, 222–223. *See also* GeoRSS; maps

#### geographical information. *See* geo-coding; locales; maps

#### GeoRSS. *See also* geo-coding

definition, 79, 224–225  
 iFrames, 230–232  
 issues, 230  
 overlays  
   BuildAmazonURI, creating, 229  
   converting Amazon data, 227–228  
   definition, 223  
   generic `getdata.aspx`, 228–229  
   XSL transformation, 225–227

#### GET method, 68, 134

#### getAllResponseHeaders ( ) method, 168

#### getdata.aspx, 161–162, 172, 228–229

#### getResponseHeader ( ) method, 168

#### getting contents of, 200–201

#### Google

AdSense API, 36  
 advertisement platform, 36  
 API, 267–269  
 Begin and End task methods, 272–274  
 blog hosting, 36  
 Blogger API, 36  
 DoGetCachedPage API, 267–269  
 DoGoogleSearch API, 267–269  
 DoSpellingSuggestion API, 267–269  
 Gadgets API, 36  
 interim XML document, 273  
 Maps API, 36  
 performance APIs, 267–269  
 personal home page, 36  
 sample code, 37  
 search application, example, 268–269  
 search results, converting to XML, 273  
 Timeout function, 274  
 Toolbar API, 36–37  
 URL for, 35  
 Web Search API, 37

**grants.** See **permissions**

**graphics.** See **images; photographs; YouTube**

**GUI (Graphical User Interface), mashups**

- Amazon search interface
  - accurate XPath, 143
  - converting the Yahoo query, 141–142
  - default namespace, 143
  - namespaces, 143
  - performance, 144
  - templates, triggering, 143
  - XSL output type, specifying, 143
  - XSL style sheet, applying, 147–149
  - XSL style sheet, description, 144–147
  - XSL transformations, 142–144
- footers, defining, 132–133
- formatting alternate rows, 132–133
- headings, defining, 132–133
- item separators, 132–133
- search results, displaying, 131–133
- templates, 132–133
- Yahoo search interface
  - generic helper class, 137–139
  - HTML, retrieving, 134–137
  - HTTP GET request, 134
  - mini-browser, C#, 135–137
  - results, sample, 134
  - serialization, 139–141
  - XML, retrieving, 137–139
  - XML documents, converting to a DataSet, 139–141
  - Yahoo REST URI example, 134–137

**GUI (Graphical User Interface), remix example, 98–102**

guid, RSS element, 115

## H

**hand-held devices.** See *specific devices*

**hCards, 83–84**

HeaderTemplate, 132

Help operation, 17

**hierarchies of products.** See *BrowseNodes*

**Historical Pricing, 25**

**HMAC (hash-based message authentication code), 310**

HMAC, storing, 201–203

**HMACSHA1 algorithm, 309–310**

home page, Google, 36

**HTTP (HyperText Transfer Protocol)**

- connectionless, 68
- definition, 67–68
- GET method, 68, 134
- headers, browsers requests, 168–169
- POST method, 68
- querystring, 68
- requesting data, 68

**HTML, retrieving, 134–137**

**human intervention, 24, 319**

## I

**identifiers, ECS FAQ, 340–342**

**iFrames, 88, 230–232**

**if-then-else statements, 165**

**illustrations.** See **images; photographs**

**Image Search, Yahoo, 31**

**images.** See *also* **photographs; YouTube**

- collages, 325–329
- display order, 217–219
- Image Search, Yahoo, 31
- ResponseGroups, 21

**individual developers, 9**

**instant messaging, 42–43**

**interface types, ECS, 14–15**

ItemLookup operation, 16

**items, RSS, 113**

ItemSearch operation, 16, 20–21

ItemSearch request, 105–108

ItemTemplate, 132

**iterating through search results, 107–108**

**iterative development, 6–7**

## J

**JAH (Just Asynchronous HTML), 158**

**JavaScript**

- arrays, 194–195
- objects, specifying, 79–80. See *also* JSON (JavaScript Object Notation)
- requester, 194

**JSON (JavaScript Object Notation)**

- definition, 79–80
- mashing with
  - architecture, 93
  - functional description, 93–94
  - pros and cons, 94
  - when to use, 94
- pros and cons, 94

**JSON Amazon customer browser**

- architecture, 236
- Dynamic Script method, 238–240
- functional description, 236–238
- Hello World example, 240–241
- notation, 242
- public APIs
  - advantages of, 243
  - versus Ajax, 243
  - creating the browser, 244–245
  - getting from Amazon, 243–245
  - processing the JSON object, 250–252
  - searching for customers, 245–247
  - testing, 252–253
  - XSL file, 247–250

**K****Kempelen, Wolfgang von, 24****L****line drawings, maps, 213**

link, **RSS element, 115**

**links**

- classifying, 32–33
- storing, 32–33
- tagging, 32–33

**list ID, 184****list operations, 16****list pages, creating, 294–297**

ListId **parameter, 184**

**listing**

- bucket contents, 315–316
- buckets, 311–312

ListLookup **operation, 16**

ListMinimum, ResponseGroup, **21**

**lists**

- owner name, specifying, 184
- searching for, 16. *See also* wish lists
- type, specifying, 184

ListSearch **operation, 16**

ListType **parameter, 184**

**Live Search, 206****local government sites, 45****locales. *See also* geo-coding; maps**

- default, 15
- definition, 15

- ECS (E-Commerce Service), 15
- switching, 104–105, 122–124

**logging, 216****look-and-feel**

- See* CSS (Cascading Style Sheets)
- See* Master Pages
- See* skins
- See* themes

**Lorem Ipsum, 159****M****machine-readable information, publishing, 81****mandatory fields, RSS, 115****maps. *See also* geo-coding; locales**

- annotations. *See* overlays
- Google Maps API, 36
- line drawings, 213
- markers, 214–216
- restaurant finder, 233. *See also* restaurant finder,
  - Yahoo maps API
- sample, 211–212
- satellite images, 41–42
- satellite photography, 213
- TerraServer USA, 41–42
- types of, 213
- Yahoo Maps, 29–30. *See also* Yahoo maps API
- zoom levels, 213–215

**Maps, Yahoo, 29–30****Maps API, Google, 36****markers, maps, 214–216****mashing**

- with Ajax
  - architecture, 90
  - functional description, 90–91
  - pros and cons, 91–92
  - when to use, 92
- with JSON
  - architecture, 93
  - functional description, 93–94
  - pros and cons, 94
  - when to use, 94
- on the Web server
  - architecture, 88
  - functional description, 88–89
  - iFrames, 88
  - pros and cons, 89
  - when to use, 89

**mashups, 76–77. See also remix**

**mashups, creating**

- Amazon search interface
  - accurate XPath, 143
  - converting the Yahoo query, 141–142
  - default namespace, 143
  - namespaces, 143
  - performance, 144
  - templates, triggering, 143
  - XSL output type, specifying, 143
  - XSL style sheet, applying, 147–149
  - XSL style sheet, description, 144–147
  - XSL transformations, 142–144
- approach to, 131
- architecture, 129–130
- binding data to controls, 132–133
- functional description, 130
- mashing search results, 149–152
- Repeater control, 131–133
- user interface
  - footers, defining, 132–133
  - formatting alternate rows, 132–133
  - headings, defining, 132–133
  - item separators, 132–133
  - search results, displaying, 131–133
  - templates, 132–133
- Yahoo search interface
  - generic helper class, 137–139
  - HTML, retrieving, 134–137
  - HTTP GET request, 134
  - mini-browser, C#, 135–137
  - results, sample, 134
  - serialization, 139–141
  - XML, retrieving, 137–139
  - XML documents, converting to a DataSet, 139–141
  - Yahoo REST URI example, 134–137

**mashups, performance. See also PageAsyncTasks**

- A9.com, registering with, 279–280
- Amazon, Begin and End tasks, 274
- architecture, 256
- Async client, creating, 262–267
- asynchronous techniques, 258–260
- delegates, 258–260
- functional description, 256–258
- Google API, 267–269
- Google search application, example, 268–269
- MethodAsync method, 260–267
- OpenSearch, 270

- OpenSearch Manifest file, 279–280
- parallel processing, 260–267. *See also*
  - PageAsyncTasks
- PreRenderComplete event, 275–276
- search results
  - combining, 275–276
  - pagination, 277–278
- test Web service, creating, 260–267
- testing the application, 278–279

**Master Pages, 55–57. See also CSS (Cascading Style Sheets); skins; themes**

**Mechanical Turk, 24, 319**

Menu **control, 61–62**

**message digests, 309–310**

MethodAsync **method, 260–267**

**microformats, 83–84**

**Microsoft Express Editions**

- database development, 50
- SQL Server 2005 Express Edition, 50
- URL for, 49
- Visual Basic 2005 Express Edition, 50–51
- Visual C++ 2005 Express Edition, 50–51
- Visual C# 2005 Express Edition, 50–51
- Visual J# 2005 Express Edition, 51
- Visual Web Developer 2005 Express edition, 50
- Web development, 50
- Windows development, 50–51

**Microsoft gadgets, 335–336**

**Microsoft portals, 335–336**

**MIME types, configuring, 290–294**

MimeType, **overriding, 169**

**mini-browser, C#, 135–137**

**mobile devices. See specific devices**

**monitoring blogs, 33–34**

**movies. See YouTube**

**MSN Messenger Activities, 42–43**

**MSN Search, 39**

**multiple delivery channels, 7**

**music**

- podcasts, 332
- providers, Rhapsody, 43
- Yahoo Audio Search, 30

## N

Name **parameter, 184**

Name **property, 19**

**named templates, 175–176**

**namespaces**

- default, 143
- issues, 176
- Xpath, 143

**network proxies, 164**

New Releases, ResponseGroup, **21**

**news headings, visualizing, 321–324****News Search, Yahoo, 31****O****objects**

- S3
  - definition, 306–307
  - deleting, 316–318
  - uploading, 313–314
- serialization, 75. *See also* REST (Representational State Transfer); SOAP (Simple Object Access Protocol)
- online storage. *See also* S3 (Simple Storage Service)**
  - architecture, 304
  - authentication, 308
  - AWSAccessKeyId parameter, 307–310
  - buckets
    - creating, 310–311
    - definition, 306
    - deleting, 312
    - listing, 311–312
    - listing contents of, 315–316
  - digests, 309–310
  - EC2 (Elastic Computing Cloud), 319
  - functional description, 304–305
  - HMAC (hash-based message authentication code), 310
  - HMACSHA1 algorithm, 309–310
  - message digests, 309–310
  - objects
    - definition, 306–307
    - deleting, 316–318
    - uploading, 313–314
  - permissions, 314–315
  - registering for, 306
  - required parameters, 307–310
  - secret keys, 307–308
  - signatures, 307–308, 309
  - timestamp, 308–309
  - usage, tracking, 318–319
- open ( ) **method, 168**
- OpenSearch, 270**
- OpenSearch Manifest file, 279–280**

Operation **parameter, 19–21, 184**

**OPML (Outline Processor Markup Language), 78–79****overlays**

- BuildAmazonURI, creating, 229
- converting Amazon data, 227–228
- definition, 223
- generic `getdata.aspx`, 228–229
- XSL transformation, 225–227
- overrideMimeType ( ) **method, 168**
- overriding MimeType, 169**

**P**

PageAsyncTasks. *See also* mashups, performance

definition, 270

Google

- Begin and End task methods, 272–274
- interim XML document, 273
- search results, converting to XML, 273
- Timeout function, 274
- registering, 271
- RSS generator, creating, 270–271
- SOAP proxy, begin and end, 272

**pages, WML**

- description, 287–289
- detail page, creating, 297–299
- home page, 290–294
- list page, creating, 294–297
- overview, 81–82
- sample, 82–83

**parallel processing, 260–267****parsing a tree structure, 175–176****parsing large datasets, 206**

**performance, mashups. *See also* PageAsyncTasks**

- A9.com, registering with, 279–280
- Amazon, Begin and End tasks, 274
- architecture, 256
- Async client, creating, 262–267
- asynchronous techniques, 258–260
- delegates, 258–260
- functional description, 256–258
- Google API, 267–269
- Google search application, example, 268–269
- MethodAsync method, 260–267
- OpenSearch, 270
- OpenSearch Manifest file, 279–280
- parallel processing, 260–267. *See also* PageAsyncTasks
- PreRenderComplete event, 275–276

### performance, mashups (continued)

- search results
  - combining, 275–276
  - pagination, 277–278
- test Web service, creating, 260–267
- testing the application, 278–279

### performance, XSL, 144

### permissions, S3, 314–315

### personal media centers, 6

### photographs. *See also* images; YouTube

- publishing, 34–35
- satellite images, 41–42
- sharing, 34–35

### PHP script, serialized, 80

### pictures. *See* images; photographs

### placeholder, 159

### podcasts, 332

### port assignment, browsers, 166

### portals, 335–336

### POST method, 68

### POX (Plain Old XML), 80

### PreRenderComplete event, 275–276

### privileges. *See* permissions

### product categories. *See* BrowseNodes

### Product Data data type, 14

### product database, accessing. *See* ECS (E-Commerce Service)

### product ID number, 17–18

### product operations, 16

### progress indicator, wish lists, 204–205

### proxies, network, 164

### pubdate, RSS element, 115

### publishing

- machine-readable information, 81
- photographs, 34–35

## Q

### query response, scope, 21–22. *See also* Response

#### Groups

### querystring, 68

### queue service, 24–25

## R

### RDF (Resource Description Format), 80–81

### readyState property, 171–172

### Really Simple Syndication (RSS). *See* RSS (Really Simple Syndication)

### recursion, 175–176

### recursive XSL templates, 175–176

### registering

- A9.com, 271
- for S3, 306

### remix, 76. *See also* mashups

### remix, creating

- AmazonUtility class, adding, 109–113
- application settings, 102–103
- architecture, 97–98
- configuration data, storing, 102–103
- FormatAsHTML function, 107–108
- forms
  - creating, 99–101
  - modifying, 111–112
- functional description, 98
- invoking AWS (Amazon Web Service), 105–108
- ItemSearch request, 105–108
- retrieving Amazon data, 103–105
- RSS 2.0
  - adding a feed, 121–122
  - channels, 113
  - consuming RSS data, 120–121
  - documents, sample, 113–114
  - FormatAsRSS function, 117–118
  - items, 113
  - mandatory fields, 115
  - optional fields, 115
  - outputting RSS, 117–120
  - search output, sample, 119–120
  - switching locales, 122–124
  - validating RSS, 122
  - XML documents, creating, 115–120
  - XML elements, 114
- RSS generator, creating, 109–113
- search entry screen, code, 100–101
- search results
  - converting to HTML, 107–108
  - iterating through, 107–108
- SearchAmazon function, 105–106, 105–108
- searching Amazon, 105–108
- server-side code, 102–103
- SOAP proxy class, 103–104, 105–108
- switching locales, 104–105
- UDDI (Universal Description, Discovery, and Integration), 104
- user interface, 98–102
- web.config file, 102–103

**remote cart**

- adding items to, 198, 201
- CartAdd operation, 198, 201
- CartClear operation, 198, 201
- CartCreate operation, 198, 199–200
- CartGet operation, 198, 200–201
- CartId, storing, 201–203
- CartModify operation, 198
- cookies, 202–203
- creating, 198, 199–200
- ECS FAQ, 344–345
- emptying, 198, 201
- getting contents of, 198, 200–201
- HMAC, storing, 201–203
- modifying, 198
- operations, 198–201
- Shopping Cart Contents, 14
- typical sequence of events, 198

**remote cart operations, 16****Remote Procedure Call (RPC), 84**

- removeEventListener ( ) method, 168

- Repeater control, 131–133

- Request, ResponseGroup, 21

**request operation, specifying, 184****requesting data, wish lists, 194****requests, browsers**

- aborting, 168
- creating, 169
- event handlers, 168–169
- HTTP headers, 168–169
- overriding MimeType, 169
- sending, 169–170
- triggering events, 168
- XmlHttpRequest object, 168–169

**Resource Description Format (RDF), 80–81****resources**

- describing, 80–81
- ECS FAQ, 347–348

**response handler, wish lists, 196**

- ResponseGroup parameter, 184

- ResponseGroups, 21–22

**REST (Representational State Transfer). See also****wish lists**

- description, 77–76
- Yahoo query
  - generic helper class, 137–139
  - HTML, retrieving, 134–137
  - mini-browser, C#, 135–137

- results, sample, 134
- serialization, 139–141
- XML, retrieving, 137–139
- XML documents, converting to a DataSet, 139–141
- Yahoo REST URI example, 134–137

**REST support, 14–15****restaurant finder**

- animation, 233
- architecture, 210
- browser, creating, 219–220
- data access, 221
- deploying, 229–232
- functional description, 210–211
- mapping, 233. See also restaurant finder, Yahoo
  - maps API
- Yahoo geo-coding API, 222–223
- Yahoo maps API
  - borders, 217–219
  - CSS z-index attribute, 217–219
  - debugging, 216
  - images, display order, 217–219
  - line drawings, 213
  - logging, 216
  - map types, 213
  - markers, 214–216
  - sample map, 211–212
  - satellite photography, 213
  - YLog utility, 216
  - zoom levels, 213–215

**restaurant finder, GeoRSS**

- definition, 224–225
- iFrames, 230–232
- issues, 230
- overlays
  - BuildAmazonURI, creating, 229
  - converting Amazon data, 227–228
  - definition, 223
  - generic getData.aspx, 228–229
  - XSL transformation, 225–227

- Results function, 277–278

**Rhapsody, 43****rich browser experience, 7****Robocast, 332****root nodes, XML documents, 68–69****RPC (Remote Procedure Call), 84****RSS (Really Simple Syndication)**

- aggregator, 77–78
- feed URLs, 77–78

## RSS (Really Simple Syndication) (continued)

---

### **RSS (Really Simple Syndication) (continued)**

- feeds, converting to podcasts, 332
- overview, 77–78
- reader, 77–78

### **RSS 2.0**

- adding a feed, 121–122
- channels, 113
- consuming RSS data, 120–121
- documents, sample, 113–114
- `FormatAsRSS` function, 117–118
- items, 113
- mandatory fields, 115
- optional fields, 115
- outputting RSS, 117–120
- search output, sample, 119–120
- switching locales, 122–124
- validating RSS, 122
- XML documents, creating, 115–120
- XML elements, 114

### **RSS generator, creating, 109–113, 270–271**

## **S**

### **S3 (Simple Storage Service)**

- architecture, 304
- authentication, 308
- `AWSAccessKeyId` parameter, 307–310
- buckets
  - creating, 310–311
  - definition, 306
  - deleting, 312
  - listing, 311–312
  - listing contents of, 315–316
- definition, 25–26, 303
- digests, 309–310
- EC2 (Elastic Computing Cloud), 319
- e-mail attachment manager, 336–337
- functional description, 304–305
- HMAC (hash-based message authentication code), 310
- HMACSHA1 algorithm, 309–310
- human intervention, 319
- Mechanical Turk, 319
- message digests, 309–310
- objects
  - definition, 306–307
  - deleting, 316–318
  - uploading, 313–314

- permissions, 314–315
- registering for, 306
- required parameters, 307–310
- secret keys, 307–308
- signatures, 307–308, 309
- timestamp, 308–309
- usage, tracking, 318–319

### **sandbox, 189**

#### **sandbox keys, 189**

#### **sandbox test user account, 189**

#### **satellite photography, maps, 213**

#### **scalable computing power, 25**

#### **schemas, XML, 69**

#### **Search, Yahoo, 30–31**

#### **search entry screen, code, 100–101**

#### **search page, creating, 290–294**

#### **search platforms. See also Google**

- 411Sync, 39–40

- Alexa, 26–28

- Amazon search interface

  - accurate XPath, 143

  - converting the Yahoo query, 141–142

  - default namespace, 143

  - namespaces, 143

  - performance, 144

  - real time results, 330–332

  - templates, triggering, 143

  - XSL output type, specifying, 143

  - XSL style sheet, applying, 147–149

  - XSL style sheet, description, 144–147

  - XSL transformations, 142–144

  - Zuggest, 330–332

- MSN Search, 39

- Yahoo Search

  - description, 30–31

  - generic helper class, 137–139

  - HTML, retrieving, 134–137

  - HTTP `GET` request, 134

  - mini-browser, C#, 135–137

  - results, sample, 134

  - serialization, 139–141

  - XML, retrieving, 137–139

  - XML documents, converting to a

    - `DataSet`, 139–141

  - Yahoo REST URI example, 134–137

- `SearchAmazon` function, 105–106, 105–108

**searching**

## Amazon

- FormatAsHTML function, 107–108
- SearchAmazon function, 105–106, 105–108
- SOAP proxy, 105–108
- for wish lists, 184–186, 187–188

## BrowseNodes, 17

from a cell phone, 39–40

combining results. See mashups

customer content, 16

customer transactions, 17

ECS FAQ, 345–347

lists, 16

OpenSearch, 270

OpenSearch Manifest file, 279–280

output, RSS sample, 119–120

products, 16

query response, scope, 21–22. See also Response

## Groups

results

- combining, 275–276

- converting to HTML, 107–108

- iterating through, 107–108

- pagination, 277–278

- third-party merchants, 17

**Second Life, 43–45****secret keys, 307–308****security**

- browsers, 163–164

- malicious scripts, 95

- validating returned data, 95

- wish lists, 205–206

**Seller Information data type, 14**

SellerListingLookup **operation, 17**

SellerListingSearch **operation, 17**

SellerLookup **operation, 17**

**semantic Web, 80–81**

send ( ) **method, 168**

**sending browsers requests, 169–170**

SeparatorTemplate, **132**

**serialization**

- objects, 75

- PHP script, 80

**serialized PHP, 80**

Service **parameter, 19–21**

setRequestHeader ( ) **method, 168**

**Shopping, Yahoo, 32**

**shopping cart. See remote cart**

**Shopping Cart Contents data type, 14**

**signatures, S3, 307–308, 309**

SimilarityLookup **operation, 16**

**Simple Storage Service (S3). See S3 (Simple Storage Service)**

**skin files, 65**

**skins, 63–65. See also CSS (Cascading Style Sheets); Master Pages; themes**

**SOAP (Simple Object Access Protocol), 75**

- proxy, begin and end, 272

- proxy class, 103–104, 105–108

- retrieving Amazon data, 103–105

- support, 14–15

**social networking, 8–9**

**software, as a service, 5–6**

source, **RSS element, 115**

**stateless methods, 75**

**storing data, 32–33. See also S3 (Simple Storage Service)**

**style sheets, adding, 62–63**

**styles. See CSS (Cascading Style Sheets); Master Pages; skins; themes**

**syndicating data. See data, syndication**

**T****tagging**

- links, 32–33

- news feeds, 321–324

**TagLines, 321–324**

**Technorati, 33–34**

**templates**

- matching to multiple elements, 165

- named, 175–176

- triggering, 143

- XSL, recursive, 175–176

**themes, 63–65. See also CSS (Cascading Style Sheets); Master Pages; skins**

**third-party merchants, 17**

**Third-Party Product Information data type, 14**

**thumbnails, of Web sites, 27–28**

Timeout **function, 274**

**timestamp, 308–309**

title, **RSS element, 115**

**Toolbar API, Google, 36–37**

**Top Sites feature, 27**

TopSellers, ResponseGroup, **21**

**Traffic Alerts, Yahoo, 31**

**traffic analysis, 26–28**

TransactionLookup **operation, 17**

**transformation function, 196–197**

**transforming XML documents**

Amazon search interface, 142–144

description, 71–73

GeoRSS overlays, 225–227

restaurant finder example, 225–227

YouTube responses, 165–168

**tree structure, parsing, 175–176**

**Turing, Alan, 334**

**Turing Test, 334**

## U

**UDDI (Universal Description, Discovery, and Integration), 104**

**URIs (Uniform Resource Identifiers), 67**

**user interface, mashups**

Amazon search interface

accurate XPath, 143

converting the Yahoo query, 141–142

default namespace, 143

namespaces, 143

performance, 144

templates, triggering, 143

XSL output type, specifying, 143

XSL style sheet, applying, 147–149

XSL style sheet, description, 144–147

XSL transformations, 142–144

footers, defining, 132–133

formatting alternate rows, 132–133

headings, defining, 132–133

item separators, 132–133

search results, displaying, 131–133

templates, 132–133

Yahoo search interface

generic helper class, 137–139

HTML, retrieving, 134–137

HTTP GET request, 134

mini-browser, C#, 135–137

results, sample, 134

serialization, 139–141

XML, retrieving, 137–139

XML documents, converting to a DataSet, 139–141

Yahoo REST URI example, 134–137

**user interface, remix example, 98–102**

**user tracking, wish lists, 206**

## V

**validating**

returned data, 95

RSS, 122

**Video Search, Yahoo, 31**

**videos. See YouTube**

**virtual reality, 43–45**

## W

**WAP (Wireless Access Protocol), 285**

**Web 2.0**

and Amazon, 9–11

as business platform, 4

convergence, 6

“data is king,” 6

definition, 3

design cycles, 6–7

folksonomies, 4–5

individual developers, 9

iterative development, 6–7

multiple delivery channels, 7

personal media centers, 6

rich browser experience, 7

social networking, 8–9

software as a service, 5–6

**Web browsers. See browsers**

**Web pages**

analysis, 26–28

appearance

See CSS (Cascading Style Sheets)

See Master Pages

See skins

See themes

information, getting, 28

navigation

ASP.NET Sitemap, 60–61

controls, 61–62

menus, adding, 61–62

SiteMapPath control, 61–62

ranking, 27

sample, 51–53

screenshots of, 27–28

thumbnails of, 27–28

Top Sites feature, 27

**Web Search, Yahoo, 31****Web Search API, Google, 37****Web server**

- mashing with
  - architecture, 88
  - functional description, 88–89
  - iFrames, 88
  - pros and cons, 89
  - when to use, 89

**Web services. See also WSDL (Web Services Description Language)**

- definition, 73

**Web Services Description Language (WSDL), 74****Web sites**

- analysis, 26–28
- appearance
  - See CSS (Cascading Style Sheets)
  - See Master Pages
  - See skins
  - See themes
- information, getting, 28
- navigation
  - ASP.NET Sitemap, 60–61
  - controls, 61–62
  - menus, adding, 61–62
  - SiteMapPath control, 61–62
- ranking, 27
- sample, 51–53
- screenshots of, 27–28
- thumbnails of, 27–28
- Top Sites feature, 27

**web.config file, 102–103****well-formed XML documents, 68–69****Wireless Access Protocol (WAP), 285****Wireless Markup Language (WML). See WML (Wireless Markup Language)****wish lists**

- Ajax
  - definition, 179
  - libraries, 206–207
  - pros and cons, 182
- architecture, 180
- auto-completion, 206
- browser caching, breaking, 204
- browser support, 206
- BuildAmazonURI function, modifying, 187–188
- data provider, modifying, 186
- debugging, 205

- eBay API, 189
- eBay REST interface, 190
- functional description, 180–181
- JavaScript arrays, 194–195
- JavaScript requester, 194
- list ID, 184
- list operations, 16
- list owner name, specifying, 184
- list type, specifying, 184
- ListId parameter, 184
- ListType parameter, 184
- Live Search, 206
- Name parameter, 184
- Operation parameter, 184
- parsing large datasets, 206
- progress indicator, 204–205
- project setup, 182–183
- remote cart
  - adding items to, 198, 201
  - CartAdd operation, 198, 201
  - CartClear operation, 198, 201
  - CartCreate operation, 198, 199–200
  - CartGet operation, 198, 200–201
  - CartId, storing, 201–203
  - CartModify operation, 198
- cookies, 202–203
  - creating, 198, 199–200
  - emptying, 198, 201
  - getting contents of, 198, 200–201
  - HMAC, storing, 201–203
  - modifying, 198
  - operations, 198–201
  - typical sequence of events, 198
- request operation, specifying, 184
- requesting data, 194
- response handler, 196
- ResponseGroup parameter, 184
- sandbox, 189
  - sandbox keys, 189
  - sandbox test user account, 189
- searching Amazon, 184–186, 187–188
- security considerations, 205–206
- transformation function, 196–197
- transforming list XML structure, 197
- transforming XML documents, 190–193
- user tracking, 206
- XML, sending from `getdata.aspx`, 186–187
- XSL arrays, setting up, 194–195

## WML (Wireless Markup Language)

- cards, 81–82
- decks, 81–82
- definition, 81
- detail pages, creating, 297–299
- forms, 287–289
- home page, 290–294
- list pages, creating, 294–297
- page sample, 82–83
- pages, 81–82, 287–289
- transforming Amazon data into, 289–290

## WSDL (Web Services Description Language), 74

## X

### XHTML (Extensible Hypertext Markup Language), 69–70

#### XML (eXtensible Markup Language)

- case sensitivity, 60
- data, retrieving, 137–139
- data presentation, 57–59
- definition, 68
- DTD (Document Type Definition), 69
- elements in RSS, 114
- root nodes, 68–69
- schemas, 69
- sending from `getData.aspx`, 186–187
- well-formed, 68–69

#### XML documents

- converting to a `DataSet`, 139–141
- elements, 68–69
- root nodes, 68–69
- sample, 68–69
- transforming
  - Amazon search interface, 142–144
  - description, 71–73
  - GeoRSS overlays, 225–227
  - restaurant finder example, 225–227
  - YouTube responses, 165–168
- well-formed, 68–69

`XmlHttpRequest` object, 168–169

#### XML-RPC, 84

#### XPath (XML Path Language), 70–71, 143

#### XSL (eXtensible Stylesheet Language)

- arrays, setting up, 194–195
- description, 71–73
- output type, specifying, 143
- style sheets, 144–149

tools

- `if-then-else` statements, 165
- templates, matching to multiple elements, 165
- `XmlHttpRequest` object, 168–169
- YouTube response, transforming, 165–168

## XSLT (XSL Transformations), 71–73

## Y

### Yahoo

- Amazon search. See mashups, creating;  
restaurant finder
- Audio Search, 30
- Content Analysis, 31
- geo-coding API, 222–223
- Image Search, 31
- Maps, 29–30. See also restaurant finder, Yahoo  
maps API
- News Search, 31
- Search, 30–31
- Shopping, 32
- Traffic Alerts, 31
- Video Search, 31
- Web Search, 31

#### Yahoo maps API

- borders, 217–219
- CSS `z-index` attribute, 217–219
- debugging, 216
- images, display order, 217–219
- line drawings, 213
- logging, 216
- map types, 213
- markers
  - custom, 214–215
  - tying to events, 215–216
- sample map, 211–212
- satellite photography, 213
- `YLog` utility, 216
- zoom levels, 213–215

#### Yahoo REST URI example, 134–137

#### Yahoo Search, 30–31

#### Yahoo search interface

- generic helper class, 137–139
- HTML, retrieving, 134–137
- HTTP `GET` request, 134
- mini-browser, C#, 135–137
- results, sample, 134
- serialization, 139–141

XML, retrieving, 137–139  
XML documents, converting to a DataSet,  
139–141  
Yahoo REST URI example, 134–137  
**Yahoo Term Extraction API, 322–324**  
YAHOO\_MAP\_HYB **maps, 213**  
YAHOO\_MAP\_REG **maps, 213**  
YAHOO\_MAP\_SAT **maps, 213**  
YLog **utility, 216**  
**YouTube. See also browsers; images; photographs**  
API, 160–162  
description, 40–41

response  
sample, 166–167  
transforming, 165–168  
response, transforming, 165–168  
XSL template, 167–168

## Z

z-index **attribute, 217–219**  
**Zollage, 325–329**  
**zoom levels, maps, 213–215**  
**Zuggest, 330–332**

