

Symbols

`$wnd.__dispatchEvent()` 200
`$wnd_dispatchEvent()` 199
* wildcard 549
`@typeArgs` 359, 362–363

A

About 133–134
About menu, automatic
 generation of 472
AbsolutePanel 171
 `add(Widget, x, y)` 171
 `getWidgetLeft()` 171
 `getWidgetTop()` 171
 `setPosition(Widget, x, y)` 171
abstract class 133
AbstractAsyncHandler 389
AbstractImagePrototype 130
`add()` 160–161
`add(Widget,String)` 172
`addChangeListener` 259
`addDashboardComponent()`
 190
`addMenu()` 275
Address Book 136, 218, 232
 handling scrolling 225
 handling tab events 232
`addRowToGrid()` 380
`addStyleName()` method 85
adopting widgets 184
Ajax 111, 113
 with RequestBuilder 410–424
AlphaImageLoader filter 149

anonymous classes, reason to
 avoid 423
Ant 48, 99
ant flag 48
Apache 99
Apache Foundation 462
Apache Tomcat 548
 directory structure 550
 servlet engine 87
API, exposing 278, 294
Appl-i18n tool 45
application
 altering for locale 518–520
 compiling 94–98
 constructing 72–83
 creating 44, 50, 54–63
 cross-script 572
 defining server
 resources 326–327
 deploying 98–99, 539–553
 developing 67–86
 executing in debug mode 92
 implementing
 internationalization 56–57
 internationalization 54–56
 large 67
 legacy 565
 loading mechanism 564–572
 logging 100–103
 manaing GTP
 properties 327–332
 running in web mode 99
 setting entry point 337
 standard 566–567
 styling 83–86

 testing and debugging in
 hosted mode 86–94
 application-compile script 62
 applicationCreator tool 30, 44,
 46, 48, 50, 53, 62, 353, 543
 application-loading
 mechanism 564–572
 application-shell script 62
 architecture 376
 ArrayList 362
 assertion methods 533
 assignable type 475
 AsyncCallback interface
 369, 384–385, 388–391
 asynchronous code
 testing 537–539
 asynchronous communication
 348, 350–351
 asynchronous mode 537
 asynchronous service
 interface 367, 371
 asynchronous testing 537–539
 automatic updates 394

B

bin directory 550
Black Hat convention 352
blacklist argument 88
`_blank` frame name 421, 430
blocking 392
 comparison to polling 398
 implementing on client 399
 implementing on server
 399–401
 server threads 398–401

Book application 125, 235
 managing tree events 235
 bookmarks 417
 Bookmarks menu
 designing 417–418
 XML data file 418–419
 boolean readBoolean() 406
 booleanValue() 447
 bootstrapping, standard
 applications 567–572
 breakpoints, setting 92
 browser
 change title of 128
 communicating via GWT
 Java 291–292
 communicating via
 JavaScript 292–293
 detection of type 560
 determining which is being
 used 330
 managing differences
 496–502
 tab index 133
 browser events, handling
 standard 216–231
 browser functionality,
 accessing 278
 browser-specific code 328
 builder, creating 414
 Button 35, 248, 252, 427, 454
 click() 138
 GWT Java class 112
 Java object 112
 <button>, DOM element 114
 Button widget 78, 80, 137–138
 ButtonPanel 186
 creating 183–185
 byte readByte() 406

C

cache.html 563
 cache.xml files 561–563
 identification of browser
 choices 563
 Calculator 138, 140, 221, 223
 handling clicks 221
 managing keyboard
 events 223

callbacks
 chaining 386–388
 handler 373, 416
 mechanism 385
 object, creating 369–370
 routine, using Command
 Pattern 384–391
 cancelLabelChange() 262
 canProcess() 400
 capturing
 long text 433
 text 432
 Cascading Style Sheets
 (CSS) 42, 80, 85, 355
 class-naming convention 268
 styling widgets 117
 CellFormatter 179
 CellPanel 176
 HasHorizontalAlignment 177
 HasVerticalAlignment 177
 CGI 410
 CGI (Perl module) 466
 chaining callbacks 386–388
 ChangeListener
 136, 218–221, 259
 addChangeListener() 265
 fireChange() 272
 onChange() 219
 ChangeListenerCollection
 259, 265
 changeTextLabel() 263
 char readChar() 406
 Character.isDigit() 141
 characters, escaping 412
 CheckBox 138
 for boolean values 433–434
 class replacement, DOM 332
 classes
 abstract 390
 locale-specific 519–520
 schedulable 395–398
 classpath
 confusion with GWT
 modules 322
 relation to GWT module
 file 83
 ClickListener 36, 212–213,
 221–222, 252, 454, 460
 addClickListener() 213
 removeClickListener() 213

ClickListenerCollection 213
 fireClick() 213
 client code, structuring 376–391
 client package 364
 client, implementing
 blocking 399
 client-pull 393
 client-side RPC architecture 375
 code
 compiling 94–98
 deploying 98–99
 ColourPicker 270
 constructing 273–274
 com.google.gwt.http.HTTP 455
 Comet 393
 Command 122
 execute() 122
 command class 388
 Command interface
 81, 419–420
 command object 385
 Command pattern 122, 376,
 384, 386–391
 commons-fileupload library 436
 commons-httpclient library 461
 compilation 94
 -gen 96
 GWT 556–564
 GWT, file mapping 557
 GWT, output 559
 heap size errors 97
 -logLevel 95
 -out 96
 output information in tree
 format 95
 setting level of logging 95
 setting location for generated
 code 96
 setting location of output
 code 96
 setting the style of output
 code 96
 speeding up the process of 97
 -style 96
 -treeLogger 95
 viewing results 97–98
 compilation process
 benefits of 561
 drawbacks of 561
 compilation script 95

- compiler 6
 - flags 573
 - making output more readable 573
 - web mode 573
 - compiling 94–98
 - complex panels 162
 - complexity, reducing 382
 - ComplexPanel 170–178
 - component shell, writing 377–380
 - components
 - continuously updated 393–398
 - labels, adding 380–382
 - shell, writing 377–380
 - style, adding 380–382
 - Composite 378
 - initWidget() 248
 - composite panels 162, 181–182
 - composite widgets 247, 251
 - building 258–265
 - creating 262
 - creating from other composite widgets 269–274
 - definition 247
 - development steps 248–249
 - initializing 263
 - initWidget() 263
 - layout 253
 - setWidget(), deprecated 263
 - styling 267–269
 - conf directory 550
 - configuration 14
 - configuring production environment 550–552
 - confirmDelete 190
 - constants
 - avoiding writing directly 503
 - moving to resource file 503
 - referring to as string literals 503
 - Constants interface 14
 - containsKey() 450
 - content delivery
 - client-pull 393
 - server-push 392–393
 - techniques 392
 - content-type 415
 - Content-type header 414
 - continuously updated
 - component 393–398
 - convention 388
 - CPAN 467
 - createConfirmButton() 262
 - createImage() 130
 - createMessages flag 56
 - createOptionsMenu() 488
 - creation tools 39
 - application creator 44, 48, 50, 53, 62
 - i18nCreator 511
 - internationalization
 - creator 45, 54–55, 62
 - project creator 44, 47–48, 50
 - unit testing 45, 57–58, 62
 - cross-script applications 572
 - cross-site request forgery 351
 - CSS. *See* Cascading Style Sheets (CSS)
 - custom field serializer 402–407
 - class rules 403
 - reasons to write 402
 - CustomFieldSerializer 403
-
- ## D
- Dashboard.html 558
 - Dashboard 50–53, 74, 95, 97, 103, 522
 - About 133–134
 - About menu 472
 - adding JUnit tests 57
 - Address Book 135–136, 168, 206, 218, 225, 232
 - Book 125, 235
 - bookmark menu 124
 - Calculator 138, 140, 167, 181, 221, 223
 - Colour Picker 269
 - compiling the code 94–98
 - component applications 65–66, 81
 - confirmDelete 297
 - creating panel for 187–190
 - Dashboard_APITest.html 296
 - Dashboard_intranet
 - class 123, 523
 - DashboardComposite 274–276, 334, 477
 - DashboardContainer 188
 - DashboardPanel 204, 208, 215, 222, 274, 298, 477
 - debugging 92–94
 - deploying the code 98–99
 - description 65–67
 - developing your application 67–86
 - directory structure 51, 59
 - drag and drop 65
 - EditableLabel
 - 131, 247–249, 253
 - building 250, 258–269
 - change text notification 220
 - choosing panel layout and structure 253–255
 - creating 259–260, 263
 - FlowPanel 173
 - identifying
 - components 251–253
 - implementing GWT Java interfaces 255–258
 - multiple text lines 140
 - styling 267–269
- externalvisibility
 - property 123, 521, 563, 569
- final version 522–523
- Finance News 518
- Flash movie widget 496
- FlashMovieImpl 499
- generator 477–492
- getConfirmDelete() 298
- getDashboardName() 296
- Google Video Search 308, 336
- importing into Eclipse 60
- internationalization 54–57, 68–72
- internet view 520
- intranet view 520
- key functionality 65
- languages 54
- Login 140, 173, 177
- menu system 65, 121
- module XML file 337–339
- onEventPreview 209
- passing data between independent applications 297
- PNGImage 147–148
- processing XML 65

- Dashboard (*continued*)
 - running in hosted mode
 - 90–91
 - running in web mode 99
 - Search Comparison 174
 - Server Status 131, 164, 180, 232, 234, 327
 - setConfirmDelete() 298
 - setDashboardName() 295
 - Slider 270–273
 - Slideshow 119, 128, 563
 - Stage 6 99
 - testing and debugging in host mode 86–94
 - ToggleMenuItem
 - 150–151, 154–155
 - Tooltip 231
 - Trash Icon 216
 - TwoComponentMenuItem
 - 124, 150–153, 177
- Dashboard.css file 84
- Dashboard.gwt.xml file
 - 72, 81, 86
- Dashboard.html file 72–73, 86
- Dashboard.java file 72, 77
- Dashboard_APITest.html 294
- Dashboard-compile, altering classpath 340
- DashboardComposite 334
 - definition of 274–276
- DashboardCompositeGenerator 334
- DashboardConstants interface
 - file 80
- DashboardConstants.java
 - file 68, 72
- DashboardConstants.properties 69, 506
- DashboardConstants_sv.properties 507
- DashboardConstants-i18n
 - tool 55–56, 68–70, 508
- DashboardContainer 188
- DashboardMessages.properties 512
- DashboardMessages-i18n 512
- DashboardPanel 200, 274, 298
 - gaining and losing focus 222
 - ignoring event 208
 - onMouseDown 239
 - onMouseMove 240
 - onMouseUp 240
- Dashboard-shell 90
 - altering classpath 340
- Dashboard-shell script 87
- data
 - hiding with Hidden 435–436
 - pagination 180
- data objects 348
 - JSON 445–450
 - serializable 356–360
- DBIx::JSON 467
- debugging
 - hosted mode 86–94
 - using JUnit messages 532
 - using rebind decision files 542
- DeckPanel 170, 173–174
 - setVisibleWidget() 174
 - showWidget(int) 174
- decodeComponent() 412
- default application
 - HTML file 72–77
 - Java file 77–81
 - module XML file 72, 81–83
- default browser event, preventing 216
- defaultexcludes attribute 326
- deferred binding 71, 149
 - creating DOM class instances 497
 - with generators 474, 491
- DeferredCommand 272
- define-property tag 495
- defining
 - new properties 328
 - server resource 327
- deploying
 - applications 539–553
 - code 98–99
 - to a filesystem 99
 - to a web server 98
- deployment descriptor 549
 - writing 552–553
- deserialization 350, 402
 - pattern 404
- deserialize() 403–404, 406–407
- Design Patterns: Elements of Reusable Object-Oriented Software 382
- DETAILED option 90
- determining browser 330
- development lifecycle. *See* lifecycle
- DialogBox 164–166
 - creating 165
- Dictionary 15, 517
 - get(key) 517
 - getDictionary 517
- DiskFileItemFactory 438
- disowning widgets 184
- <div> 160
- div element 78
- DockPanel 175–177
- DOM
 - browsers dealing with 332
 - browser-specific methods 497
 - creating an element 144
 - div element 78
 - eventCancelBubble() 200
 - eventPreventDefault() 200
 - getElement() 144
 - getting specific named 171
 - implementation class
 - hierarchy 497
 - named 74
 - setAttribute() 144
- DOM pages, getting
 - reference 171
- DOMImplIE6 280
- DOMImplStandard 197, 280
- double readDouble() 406
- drag-and-drop
 - drag 236–241
 - drop 241–244
 - implementing 236–244
- DummyCommand 81
- dynamic string
 - internationalization 516–518

E

- Echo2 26
- Eclipse
 - importing into 60–61
 - launch configuration 51
 - launch files 534
- eclipse flag 39, 47, 51, 59, 61
- EditableLabel 140, 220
 - managing change 220
- EMBED tag 498

- encapsulating remote calls
 - 382–384
- encodeComponent() 412–413
- encoding-type 436
- endless loop, creating 400
- entry point
 - setting 337
- EntryPoint interface 34, 78
 - class 80
- errors, handling with tests
 - 532–533
- escaped values
 - decoding 412
 - encoding 412
- escaping 413
 - characters 412
 - URLs 412
- event handling, preventing
 - default in browser 216
- event listeners
 - ChangeListener 136
 - ClickListener 131, 138
 - HistoryListener 119
 - KeyboardListenerAdapter 141
 - LoadListener 127
 - MouseListener 131
 - PopupListener 124
 - TreeListener 126
- Event object 204
- eventBits attribute 114, 202
- __eventBits field 203
- EventListener 211, 214
- EventPreview 208–210
- events 193
 - addEventListener() 196–197
 - bubbling 195
 - BUTTON_LEFT 193
 - BUTTON_MIDDLE 193
 - BUTTON_RIGHT 193
 - capturing 195
 - capturing right mouse
 - clicks 206
 - ChangeListener 218–221
 - ClickListener 212
 - closing pop-ups 231–232
 - dispatchEvent() 197
 - eventCancelBubble() 200
 - eventGetAltKey 205
 - eventGetButton 205
 - eventGetClientX 205
 - eventGetClientY 205
 - eventGetCtrlKey 205
 - eventGetFromElement 205
 - eventGetKeyCode 205
 - eventGetRepeat 205
 - eventGetScreenX 205
 - eventGetScreenY 205
 - eventGetShiftKey 205
 - eventGetTarget 205
 - eventGetToElement 206
 - eventGetType() 204, 206
 - eventGetTypeString 206
 - eventPreventDefault()
 - 200, 216
 - EventPreview 208–210
 - finding a listener 199
 - FOCUSEVENTS 194
 - FocusListener 215, 275
 - form events 231
 - getting the type of 204
 - GWT event model 196–200
 - GWT handling lifecycle 197
 - handling 193, 199
 - handling by extending listener classes 211–212
 - handling double click 200
 - handling standard browser events 216–231
 - identifying event model
 - browser differences 195–196
 - KeyboardListener 140, 168
 - KEYEVENTS 194
 - linking sunk events to
 - onBrowserEvent() 207–208
 - listening 200–216
 - LoadListener 223
 - managing sunk events
 - 203–207
 - MOUSEEVENTS 194
 - MouseListener 211
 - moving into GWT code
 - 212–216
 - ONBLUR 193
 - onBrowserEvent() 197, 199,
 - 203, 207–208
 - ONCHANGE 193
 - ONCLICK 193
 - ONDBLCLICK 194
 - onDetach() 200
 - ONERROR 194
 - onError 127
 - onEventPreview() 208
 - ONFOCUS 194
 - onHistoryChange() 119
 - ONKEYDOWN 194
 - ONKEYPRESS 194
 - ONKEYUP 194
 - ONLOAD 194
 - onLoad 127
 - ONLOSECAPTURE 194
 - ONMOUSEDOWN 194
 - ONMOUSEMOVE 194
 - ONMOUSEOUT 194
 - ONMOUSEOVER 194
 - ONMOUSEUP 194
 - ONSCROLL 194
 - PopupListener 232
 - preventing previewing 188
 - previewing 208–210
 - ScrollListeners 224–226
 - sinkEvent() 197, 202
 - sinkEvents() 197, 200–203
 - sinking 200–203
 - super.onBrowserEvent() 197
 - tab events 232–233
 - table events 234
 - TableListener 178, 234
 - TabListener 181
 - tree events 235–236
 - TreeListener 235
 - unsinkEvents() 205
 - W3C DOM model 196
 - window close events 227–231
 - Window CloseListener 293
 - window resize events 226–227
 - WindowCloseListener
 - 227–231
 - WindowResizeListener
 - 226–227, 271
- examples
 - applicationCreator 30
 - Server Status 353–356
 - Tic-Tac-Toe 32
- exceptions
 - JSNI 290
 - serializable 363
 - throwing 363
- excludes 326
- execute() 420
- existing methods,
 - overwriting 484–485

exposing an API 294
 extending properties 329
 extend-property tag 495
 Extensible Messaging and Presence Protocol (XMPP) 342
 externalvisibility property 95
 extractYahooResults() 458

F

Façade pattern 376, 382–384
 failure, different than an error 532
 field serializers
 class 403–404
 custom 402–407
 custom, class rules 403
 custom, implementing 404–407
 custom, reasons to write 402
 file uploads 436
 file-input tag 425
 FileItemFactory 438
 files, uploading with
 FileUpload 436–439
 filesystem, deploying to 99
 FileUpload 118–119, 436–439
 getFilename() 119
 getName() 119
 old version 143–146
 server-side component 118
 setName() 119
 FileUploadException 438
 FileUploadServlet class 118
 filtering public resources 326
 finishTest() 538
 fireChange() 265
 Firefox Web Developer extension 73
 Flash widget
 building 498–501
 Internet Explorer 498
 Internet Explorer implementation 500–501
 other browser implementation 500
 other browsers 498
 FlexCellFormatter 180
 spanning cells and rows 180

flexibility, adding 390
 FlexTable 179–180
 FlexCellFormatter 180
 Flickr, label editing 250
 float readFloat() 407
 FlowPanel 159–160, 173, 253, 454
 getWidget(int) 173
 getWidgetCount() 173
 getWidgetIndex(Widget) 173
 remove(int widgetIndex) 173
 setting style of 263
 FocusListener 215, 222, 275
 onFocus() 275
 onLostFocus() 275
 FocusPanel 164, 167–168
 setFocus() 168
 setTabIndex() 168
 FocusWidget 133–141, 143, 167
 setAccessKey() 133
 setFocus() 133
 setTabIndex() 133
 <form> 425
 form action 426
 form controls 425–426
 form encoding 415
 form method 425
 FormHandler 428
 onSubmit() 428
 onSubmitComplete() 428
 FormPanel 164, 166–167, 410, 424–439
 events 428–429
 form controls 432–439
 setAction() 426
 setMethod() 426
 submit() 167, 427
 submitting 427
 target, altering 429–431
 FormPanel.METHOD_GET 426
 FormPanel.METHOD_POST 426
 forms, handling events 231
 FormSubmitCompleteEvent 428
 Frame 134
 frame.setURL('resource-to-load') 134
 frames, named 429

G

gen argument 90, 96
 generate() 476
 generating new code
 automatically 471–491
 GeneratorContext 478
 generators 472
 amending classpath to use 473
 basic code 476
 context argument 476
 context properties 479–480
 DashboardCompositeGenerator 334
 deferred binding with 474
 generate() 478
 generating classes 563
 GeneratorContext object 478
 input class 478–479
 internationalization 472
 introduction 472–473
 JUnit 472
 logging progress 480–482
 logging to a TreeLogger 476
 makeInterface() 483
 PrintWriter 483
 PropertyOracle object 479
 registering 333–334, 474
 returning new class 491–492
 returning null as the result 476
 RPC 472
 SourceWriter 478, 482
 TreeLogger 480
 TypeOracle 478
 generics 358
 GET 411–413
 compared to POST 413–414
 get() 171, 450
 get(elementName) 171
 getAttribute() 144
 getCellFormatter() 179
 getDescription() 290
 getElement() 114, 378
 getGSearch() 315
 getInstance() 382–383, 447
 getMetaProperty() 521
 getName() 290, 427
 getprint() 467

getResult() 428
 getRowFormatter() 179
 getStatusCode() 416
 getStatusData() 361, 382, 384, 387
 getStatusDataFromServer() 380, 385, 397
 getType() 479
 getValue() 447
 getWidgetCount() 173
 getWidgetIndex(Widget) 173
 Google Ajax Search 299, 301, 309
 API key 299
 GdrawOptions 310
 GnewsSearch 310
 GSearchControl 310
 GsearcherOptions 310
 GwebSearch 310
 KeepListener 312
 making the widget 314
 SearchControllImpl 311
 Google Groups 464
 Google Maps 300
 Google Video Search 300, 336
 GSvideoBar object 303
 injecting JavaScript source 336
 JavaScript example 303
 making the widget 306
 Google Web Toolkit. *See* GWT
 GoogIipse 63
 Grid 35, 138, 181
 GWT 4
 \$doc 290, 292
 \$wnd 290, 292, 294
 application development lifecycle 39–43
 application-loading mechanism 564–572
 browser differences 496
 compilation 556–564
 compilation process 556, 564, 573
 compilation process, file mapping 557
 compilation process, Java objects supported 574
 compilation process, knowledge of browser types 561

compilation process, output 559
 compilation, file mapping 557
 compiler 41
 components styling 83
 creating applications 39, 44–63
 creation tools 39–42, 46, 51–52, 62, 67
 default application 39, 44, 47, 50, 65, 105
 development lifecycle. *See* lifecycle
 DOM implementation class hierarchy 497
 event model 196–200
 generators 65–66, 71, 473
 getModuleBaseURL() method 553
 history functions 76
 hosted-browser 41
 integration into Eclipse 59
 internationalization support 502
 introduction 5
 local, altering the application 518
 modularization 68
 module XML 318
 online API reference 118
 open source 564
 properties 495
 properties, externalvisibility 479
 stages of application development 42
 testing and deploying 527–554
 testing code with JUnit 528–539
 user.agent property 495
 user-defined properties 520
 using IDEs other than Eclipse 62
 vs. Echo2 26
 vs. JavaServer Faces 27
 vs. Ruby on Rails 29
 vs. Swing 25
 widgets default style names 85
 <gwt:module> 565

GWT applications
 talking between 296–298
 talking to via JavaScript API 294–296
 GWT Interfaces
 HasText 144
 GWT Widget Library 147, 340
 GWT.create() 71, 80, 366, 368
 GWT.create(classLiteral) 474
 GWT.getModuleBaseURL() 369
 GWT.isScript() 369
 gwt.js 544, 558–559
 gwt.js JavaScript file 75
 GWT.log() 101–102, 370
 gwt:module meta-tag 74
 gwt:onPropertyErrorFn meta-tag 75
 gwt:property meta-tag 75
 GWTDesigner 63
 gwtOnLoad() 572, 578
 gwt-onLoadErrorFn meta-tag 75
 GWT-RPC 348, 376
 defining 360–366
 implementing 356–370
 preparing client side of call 366–370
 RPC concepts 348–353
 servlets 550
 gwt-servlet.jar 551
 GWTShellServlet 549
 GWTTestCase 22, 537
 delayTestFinish() method 537
 finishTest() method 537
 setModuleName() method 537
 gwt-user.jar 280

H

handleFailure() 389
 handler 35
 handleSuccess() 389
 HasHorizontalAlignment 256
 required methods 259
 HashSet 362
 HasName 257, 427
 HasText 143–144, 255–257
 required methods 259
 HasWidgets interface 163

- HasWordWrap 256
 - required methods 259
 - heap size 97
 - HelpMenuName key 69
 - Hidden, hiding data 435–436
 - hiding text 433
 - History 119
 - getTargetHistoryToken() 120
 - HistoryListener 119
 - including in HTML page 120
 - onHistoryChange() 119
 - setTargetHistoryToken() 120
 - history
 - history.html 564
 - storage 564
 - history files, removing 543
 - history functions 76
 - History object 21
 - history tokens 120
 - history.html 543, 558
 - History.html file 564
 - HistoryListener 21, 119
 - HorizontalPanel 170, 177, 253
 - hosted development
 - controller 91
 - hosted mode 333, 369, 573
 - default project URL 365
 - deployment descriptor 552
 - gen 90
 - logLevel 89
 - noserver 87
 - out 90
 - setting location for generated code 90
 - setting location of output code 90
 - setting the style of output code 90
 - style 90
 - testing 365–366
 - testing and debugging in 86–94
 - whitelist 87
 - hst() 564
 - HTML 13, 252
 - getHTML() 120
 - security implications 133
 - setHTML() 120
 - HTML panels 178–181
 - HTML Table panels 162
 - HTML widget 132–133
 - HTMLPanel 172
 - HTMLTable 170, 178–179
 - setting style of cells 179
 - HTTP
 - GET 411–413
 - GET, compared to POST 413–414
 - methods 411–414
 - POST 413–414
 - response codes 416
 - HTTP module
 - importing to module configuration 412
 - RequestBuilder 414
 - HttpRequest 111
 - HttpServlet 372
 - Husted, Ted 57
 - Hyperlink 119–120, 460
 - getText() 120
 - setText() 120
 - treating as HTML 120
 - treating as text 120
 - hyperlinks, creating a true hyperlink 133
- I**
-
- i18n 45, 54–56
 - adding to a structure 55
 - module XML file 319
 - I18N GWT module 82
 - i18nCreator 511
 - i18nCreator tool 45, 54–55, 62
 - IDE 59
 - Eclipse 39, 41, 44, 47, 50, 53, 55, 57–60, 62
 - Googlipse 63
 - GWTDesigner 63
 - importing into 59–63
 - importing to 45
 - IntelliJ 63
 - Matisse 59
 - NetBeans 59, 63
 - other than Eclipse 62
 - with GWT wizard 63
 - IFRAME 76, 134
 - iframes 424
 - ignore flag 53, 59
 - IllegalArgumentException 459
 - Image 127–131
 - getURL() 150
 - PNG problem with Internet Explorer 147
 - image.setVisibility(false) 224
 - ImageBundle 129
 - images
 - bundling together 128
 - loading 223–224
 - importing into IDE 59–63
 - includes 326
 - indent() 484
 - IndexedPanel 173–174
 - inheriting other modules 321–325
 - <inherits> 365
 - inherits tag 365
 - init() 196
 - initialization code 578–579
 - Initializing a Composite Widget 263
 - initWidget() 379
 - injecting a style sheet resource 336
 - injecting JavaScript code 335
 - injection 86
 - innerHTML 149
 - <input> 428
 - installing RPC servlets 548–553
 - instantiate() 403–405
 - int readInt() 407
 - IntelliJ 63
 - intercomponent
 - communication 297
 - internationalization 14, 45, 54, 65, 68–72, 97, 328, 495, 560
 - change locales 71
 - constants 504–509
 - ConstantsWithLookup 504, 509–510
 - default locale 69
 - Dictionary 517
 - dynamic string 516–518
 - English 69
 - hierarchy 504
 - implementing 56–57
 - ISO country code 502
 - ISO language code 502
 - key/value pairs 69
 - Localizable interface 504

- internationalization (*continued*)
 - Message 504
 - messages 510–512
 - messages for default locale 512–514
 - messages for other locales 514–516
 - messages structure 512
 - properties files 69
 - setting up 54–56
 - static string 503–516
 - supporting in full 502–518
 - Swedish 69–70
 - use in the Dashboard application 124
 - use of generator 472
 - InterruptedException 400
 - introspection 65, 485–491
 - in generators 334
 - JClassType 486
 - JType 479
 - iSEC Partners 352
 - ISO country code 502
 - ISO language code 502
 - IsSerializable 357
 - interface 357–358, 361, 402
 - isString() 460
 - iterator() 170
- J**
-
- Jabber 342
 - Jakarta project 461
 - Java
 - class name 53–54
 - compiling to JavaScript 573–579
 - creating package structure 343
 - package structure 318
 - packages 343
 - strong type model 280
 - Vector class 574–576
 - Java 1.4 standard 81
 - Java 1.5, use in GWT generators 476
 - Java 5.0 standard 81
 - Java classpath, confusion with GWT modules 322
 - Java debuggers 101
 - Java libraries, adding to classpath 90
 - Java method signature 289
 - Java Native Interface. *See* JNI
 - Java objects, passing across Java-to-JavaScript boundary 283–286
 - Java package
 - client 43, 50–53, 55, 57–58, 343
 - creating structure 343
 - rebind 54, 343
 - server 41, 43, 54, 343
 - Java Runtime Environment 10
 - Java Virtual Machine (JVM) 97
 - java.io.File 439
 - java.util.Set 450
 - JavaScript 278
 - \$doc 290, 292
 - \$wnd 290, 292–294, 304, 311
 - access browser functionality 278
 - call format 575
 - compiling Java to 573–579
 - document object 290, 292
 - exposing an API 278, 294
 - files, organizing 544
 - including in a GWT application 281–283
 - intercomponent communication 278, 297
 - libraries 278
 - loose type model 280
 - permutations 495, 560–561
 - permutations, cache.html 563
 - permutations, Dashboard 579
 - permutations, reducing number of 561
 - permutations, selecting 561
 - program code as 576–578
 - relocating code 544, 546–548
 - undefined object 287
 - variable 278
 - window object 290, 292, 294
 - JavaScript API, talking to GWT application 294–296
 - JavaScript libraries
 - accessing objects 302–307
 - accessing of 278, 299
 - calling callback code 312–314
 - loading 299–300
 - loading through HTML 299
 - loading through module XML 299–300
 - third-party, accessing 278
 - wrapping as GWT component 300–309
 - wrapping complex 309–315
 - JavaScript Native Interface. *See* JSNI
 - JavaScript Object Notation. *See* JSON
 - JavaScript objects, keeping as Java objects 310–312
 - JavaScript resources, injecting 335–336
 - JavaScriptException 290
 - JavaScriptObject 284, 287, 302, 304
 - GSVideoBar example 305
 - JavaServer Faces. *See* JSF
 - JBoss 548
 - JClassType.getFields() 486
 - JClassType.getMethods() 486
 - JNI 281
 - JRE emulation library 10, 358
 - JSF 27
 - JSNI 8, 190, 281
 - accessing a field in a Java object 285–286
 - accessing a static field 285–286
 - accessing an instance field 285–286
 - calling JavaScript code 282
 - communicating using 290–298
 - creating new Java objects in JavaScript 283
 - exceptions 290
 - executing methods in Java objects 287–290
 - introduction 279–290
 - JavaScriptObject 284
 - passing a byte to JavaScript 283
 - passing a char to JavaScript 283
 - passing a double to JavaScript 283

JSNI (*continued*)

- passing a float to JavaScript 283
- passing a JavaScriptObject to JavaScript 284
- passing a long to JavaScript 283
- passing a short to JavaScript 283
- passing a String to JavaScript 284
- passing a user-defined Java object to JavaScript 284
- passing an array to JavaScript 284
- passing an int to JavaScript 283
- passing Java objects to JavaScript 283
- returning a Java object 286
- returning a JavaScriptObject 287
- returning a String 286
- returning null objects 287
- returning primitive types 286
- syntax 281

JSON 111, 469

- arrays 444
- basic types 446–447
- classes, adding to configuration file 453
- creating Search component 450–460
- data format 443–444
- data objects 445–450
- deserializing data 445
- format 443
- implementations 462
- introduction 443–445
- module 453
- number 444
- Perl module 465
- response 458
- sending data to server 455–457
- server proxy 462
- server response 457–460
- using with Java on server 461–465
- using with Perl 465–467

- using with Ruby 467–469
- value types 469

json library for Ruby 467

JSON.parse() 468

JSONArray 458

- storing JSONValue objects 448–449

JSONBoolean 447

JSONException 459

JSONNull 447–448

JSONNumber 446

JSONObject 456, 459

- collecting JSONValue objects 449–450

JSONParser 443, 445

- parse() method 459

JSONString 446

JSONTokenizer 464

jsonToObj() 466

JSONValue 445–446, 458, 460

JSONValue objects

- collecting in JSONObject 449–450
- storing in JSONArray 448–449

JUnit 22, 45, 54, 57, 59

- downloading 529

JAR file 57

- overview for GWT developers 529–534
- testing GWT code 528–539
- version compatibility with GWT 529

JUnit in Action 57

junitCreator

- command-line options 534
- creating TestCase 534
- running 534–535

junitCreator tool 45, 57–58, 62

-junitflag 58

K

- keyboard events
 - eventGetKeyCode() 209
 - getting key pressed 209
- keyboard, capturing inputs 223
- KeyListener 140, 168, 223
- KeyListenerAdapter 223
- keySet() 450

L

Label 131–132, 248, 252

- adding a ClickListener 262
- adding to component 380–382
- create as plain text 131
- editing by clicking 250
- getText() 132
- setHorizontalAlignment() 131
- setText() 132
- setWordWrap() 131

Label object 78

Label widget 80

legacy applications 565

libwww-perl 465

lifecycle 39–44, 67, 94, 103

- stage 1, tools 45, 65
- stage 2, development 67–86
- stage 3, testing and debugging 86–94
- stage 4, compiling 94–98, 556
- stage 5, deployment 98–99
- stage 6, running in web mode 99
- stages 41

LinkCommand 419

- loading web page 420–421

List interface 362

list, displaying in a ListBox 434–435

ListBox 135–137, 434–435

- drop-down view 136
- getItemText() 136
- getSelectedIndex() 136
- getSelectedItem() 137
- setMultipleSelect() 137
- setSelectedIndex() 137
- setting height 136
- standard view 136

listener.onBrowserEvent() 208

listeners

- ChangeListener 218–221
- ClickListener 212, 252
- FocusListener 215
- KeyListener 140
- LoadListener 223
- MouseListener 211
- MouseListener 232
- ScrollListeners 224–226

- listeners (*continued*)
 - TableListener 234
 - TreeListener 235
 - Window CloseListener 293
 - WindowCloseListener 227–231
 - WindowResizeListener 226–227
 - loading error function, registering 571
 - LoadListener 127, 223
 - onError() 223
 - loadSubMenu() 419, 423–424
 - locale 502
 - changing 91
 - changing in HTML 506
 - changing through URL 506
 - hierarchy 502
 - permutations based on 560
 - locale-specific classes 519–520
 - Localizable interface 504
 - localization 15
 - Log4j 89, 103
 - Log4JavaScript 103
 - Log4Perl 103
 - Log4PHP 103
 - logging 100–103, 386
 - automatic 391
 - handler 389
 - in a generator 480–482
 - on the client side 100–103
 - on the server side 103
 - logging levels 89
 - LoggingAsyncHandler 391
 - Login 140
 - logLevel argument 89, 95
 - logs directory 550
 - long readLong() 407
 - LWP 465
 - LWP::Simple 467
- M**
-
- Massol, Vincent 57
 - Matisse 59
 - Maven 99
 - maxLength vs. TextBox width property 432
 - maxResults 456
 - maybeStartModule() 572
 - MD5 name, determining 571
 - menu system 65
 - MenuBar 80, 121–124
 - addItem() 122
 - onPopupClosed() 124
 - widget 80
 - MenuItem 121–124, 150–155
 - extending 146
 - getCommand() 123
 - getParentMenu() 122
 - getSubMenu() 122
 - setCommand() 123
 - setSubMenu() 122
 - treating as HTML 123
 - MenuLoaderHandler 419, 421–423
 - menus, enabling/disabling auto-opening of children 123
 - menuTitle 423
 - Messages 14
 - metaProps 569
 - metatags 74, 545
 - gwt:module 568
 - gwt:onLoadErrorFn 569, 571
 - gwt:onPropertyErrorFn 569
 - gwt:property 568–569
 - processing 568
 - property, managing 569
 - methods
 - making final 391
 - overwriting existing 484–485
 - using introspection 485–491
 - Model View Controller. *See* MVC
 - modularization 68, 319–321
 - creating 318–339
 - module configuration 550
 - importing HTTP module 412
 - module flag 59
 - module XML 81, 561
 - property replacement 501–502
 - registering a generator 474
 - replace tags 497
 - user defined property provider 521
 - module.nocache.js file 559–561
 - module.nocache-xs.js file 559, 561
 - modules
 - class replacement due to properties 332–333
 - defining properties 328
 - defining server resource 327
 - extending properties 329
 - filtering public resources 326
 - GWT Widget Library 322
 - HttpRequest 322
 - include other modules 321–325
 - injecting a style sheet 336
 - injecting JavaScript code 335
 - internationalization 322
 - JSON 322
 - packaging 341–343
 - path to public resources 325
 - property provider 330
 - registering Generator class 333–334
 - replace class due to properties 333
 - setting source code path 325–326
 - setting the entry point 337
 - standard with GWT 319
 - third-party, using 339–341
 - User 322
 - XML 322
 - XML processing 322
 - Mouse events 207
 - eventGetButton() 207
 - mouse inputs 224
 - MouseListener 211
 - onMouseDown() 211, 237
 - onMouseMove() 237
 - onMouseUp() 237
 - multipart messages 436
 - MVC 26
 - myAppConstants.properties 504
 - MyAsyncHandler 389–390
 - MyLoggingHandler 390
- N**
-
- named DOM elements 74
 - benefits 77
 - NamedFrame 134, 429, 431
 - naming convention for widget styles 268
 - native keyword 282
 - native, declaring method 9
 - .NET 410
 - Net::Http::get_response() 469
 - NetBeans 59, 63

nocache-xs.js 558
 nontransient fields 358
 noserver argument 87
 null pointer errors 94
 NullPointerException 459

O

OBFUSCATED option 90
 Object readObject() 407
 OBJECT tag 498
 objects, sending across
 JavaScript-to-Java boundary
 286–287
 onBefore 232
 onBrowserEvent()
 117, 145–146, 162, 203–208
 handling sunk events 203
 overriding 203
 onClick() 78
 onCompletion() 539
 onError() 370
 onEventPreview() 190
 onFailure() 370, 387
 onKeep() 312
 onKeyPress() 141
 onLostFocus() 240
 onModuleLoad()
 78, 80, 123, 227
 onMouseDown() 207, 239
 onMouseDown(Widget sender,
 int x, int y) 237
 onMouseEnter(Widget
 sender) 237
 onMouseLeave(Widget
 sender) 237
 onMouseMove() 240
 onMouseMove(Widget sender,
 int x, int y) 237
 onMouseUp(Widget sender,
 int x, int y) 237
 onResponseReceived() 422, 538
 onSelected() 232
 onSink() 202
 onSuccess() 370, 387
 onWindowClosed() 228
 onWindowCloseListener() 293
 org.gwtbook 364
 organization 540
 organizing projects 540–548
 out argument 90, 96

-out flag 56, 59
 outdated() 484
 -overwrite flag 53, 59
 overwriting existing
 methods 484–485

P

Package Explorer 60
 package names, restrictions 364
 package structure, creating 343
 pageYOffset 293
 Panel class 162
 Panel, Grid 138
 panels 12, 158
 AbsolutePanel 171
 add() 161, 170
 adding widgets 159
 adopt() 184
 alignment 161
 as DOM elements 160–162
 as Java objects 159–160
 CellPanel 176
 ComplexPanel 162, 170–178
 composite panel 162
 creating 182–187
 creating by extending an exist-
 ing panel 186–187
 creating for Dashboard
 187–190
 creating from scratch
 183–185
 DeckPanel 170, 173–174
 DialogBox 164–166
 difference from widgets 163
 disown() 184
 DockPanel 175–177
 extending, vs. composite
 widgets 182
 FlexTable 179–180
 FlowPanel 160, 173, 253
 FocusPanel 164, 167–168
 FormPanel 164, 166–167
 getChildren() 170
 Grid 181
 HorizontalPanel
 170, 177, 253
 HTML table panel 162
 HTMLPanel 172
 HTMLTable 178–179
 IndexedPanel 174
 insert() 161, 170
 onAttach() 163
 PopupPanel 164–165, 270
 RootPanel 158, 161, 171
 ScrollPane 164, 168–170
 setting styling 159
 simple panel, definition 162
 SimplePanel 164–170
 StackPanel 174–175
 standard GWT 162–182
 TabBar 181
 TabPanel 173, 181–182
 VerticalPanel 160–161,
 170, 177–178, 253
 _parent frame name 430
 parentMenu 423
 parkComponent 190
 parse 445
 password entry, securing 140
 PasswordTextBox 140, 433
 Perl 443
 using with JSON 465–467
 permutations 94
 reducing number of for
 testing 562
 PHP 410
 Pjax 393
 PNGImage 147–148
 onBrowserEvent() 150
 sinkEvents() 150
 PNGImageImpl 149
 PNGImageImplIE6 149
 polling 392
 comparison to blocking server
 threads 398
 issues 392–393
 techniques 392–401
 PopupListener 124, 232
 onPopupClosed() 232
 PopupPanel 164–165,
 208, 231, 270
 pop-ups, reacting to
 closing 231–232
 POST 413–414, 456
 prefetch() 128
 PRETTY option 90
 PrintWriter 483
 process() 400
 production environment,
 configuring 550–552

- program code, as
 - JavaScript 576–578
 - project
 - creating 47–50
 - importing into Eclipse 60–61
 - organizing 540–548
 - project files, separating from
 - other pages 544
 - projectCreator 44, 47–48, 50, 91, 353
 - projectCreator tool 47
 - properties 495
 - defining 328–330
 - defining initial value 496
 - defining new 328
 - definition in module
 - XML 495–496
 - determining value through
 - code 496
 - extending 328–330
 - extending in module
 - XML 495
 - managing 327–332
 - managing differences
 - 330–332
 - user defined 520–523
 - user specified 521
 - properties file 14
 - encoding 70
 - format 69
 - hierarchy 70
 - key/value separator 69
 - UTF-8 69
 - property error function
 - registering 569–571
 - property provider 330
 - establishing 567–568
 - tag 496
 - user-specified 521–522
 - property replacement 501–502
 - PropertyOracle 479–480
 - getPropertyValue() 480
 - providing code to determine
 - properties 330
 - proxy 351, 461
 - proxy classes 361
 - proxy object 366, 384
 - casting to
 - ServiceDefTarget 368–369
 - creating 368
 - proxy service 461
 - public attribute 51
 - public folder 364
 - public tag 326
 - Pushlet 393
- R**
-
- radio buttons 434
 - RadioButton 138–139, 434
 - grouping together 139
 - reader methods 406
 - rebind 476
 - rebind decision files,
 - removing 541–542
 - reducing complexity 382
 - reflection 361
 - refresh rate 393
 - registering Generator class
 - 333–334
 - releaseCapture() 237, 239
 - remote calls,
 - encapsulating 382–384
 - Remote Procedure Call. *See* RPC
 - remote server
 - calling 368–370
 - restrictions for communicat-
 - ing with 351–353
 - remote service call
 - making 370
 - RemoteService interface
 - 361, 364, 368
 - extending 361–362
 - RemoteServiceServlet
 - 363, 372, 553
 - removeChangeListener 259
 - removeMenu() 275
 - removing history files 543
 - <replace-with> 541
 - replacing class due to property
 - values 332–333
 - Request
 - cancel() method 415
 - checking status 415
 - isPending() method 415
 - setting properties 414
 - request.getParameter() 415
 - RequestBuilder 423, 439, 455
 - loading XML data 417–424
 - RPC 414–417
 - sendRequest() method 415
 - with Ajax 410–424
 - RequestBuilder.GET 414
 - RequestBuilder.POST 414
 - RequestCallback 416, 538
 - onError() method 416
 - onResponseReceived()
 - method 416
 - RequestException 415
 - request-response cycle 411, 464
 - resources, injecting at
 - runtime 334–337
 - Response
 - checking status codes 416
 - getHeader() 417
 - getHeaders() 417
 - getHeadersAsString() 417
 - getText() 417
 - ResponseCallback 456
 - Result property 453
 - ResultSet 452
 - right mouse clicks,
 - capturing 206
 - ROOT directory 551
 - RootPanel 35, 80, 158, 161, 171, 379
 - RootPanel.get() method 80
 - RootPanel.get().add(widget/
panel) 199
 - RPC 16, 94, 111, 326, 348
 - installing servlets 548–553
 - logging 391
 - underlying concepts 348–356
 - with RequestBuilder 414–417
 - Ruby 443
 - using with JSON 467–469
 - Ruby on Rails 29
 - RubyForge 467
 - RubyGems 467
 - running tests 531–532
 - runtime errors 532
 - RuntimeException 150, 262
- S**
-
- Scalar Vector Graphics 13
 - schedulable class 395–398
 - schedule() 395
 - scheduleRepeating() 395
 - scheduling tasks 392–401
 - script.aculo.us 252
 - Scriptaculous 341
 - use in the Dashboard 341

- scrolling 168
- ScrollListeners 224–226
- ScrollPane 164, 168–170
 - ensureVisible() 169
 - setAlwaysShowScrollBars() 169
 - setHorizontalPosition() 169
 - setScrollPosition() 169
- search component, creating with JSON 450–460
- SearchResultsHandler 456
- searchString 456
- security 351
- _self 420, 429
- sendRequest() 424
- serializable data objects 371
 - following rules for client-side code 358
- SerializableException 363
- serialization 348, 402, 405
 - custom, implementing 404–407
 - types supported by GWT 357
- SerializationException 404–406
- SerializationStreamReader 404–406
- SerializationStreamWriter 405
- serialize() 403, 405–406
- serializers
 - class 403–404
 - custom 402–407
 - custom, class rules 403
 - custom, reasons to write 402
- server
 - contacting 348
 - implementing blocking 399–401
- server package 365
- server proxy 462
- server resources, defining for an application 326–327
- ServerStatus 234, 348, 552
 - creating 353–356
 - encapsulating 376–382
 - encapsulation 376
 - updating 393
- server threads
 - blocking 398–401
- server-push 351, 392–393
- emulating 398–401
- unavailable to browsers 398
- ServerService 363, 382–383
- server-side 4–6, 23
 - classes 372
 - code 364
 - GWT-RPC 99, 360
 - Java compliance 81, 415
 - JSON 450
 - logging information 103
 - service implementation 363–364, 371
 - successful response 416
 - testing functions 537–539
 - throw exception 370
- ServerStatsUpdater 385, 390
- ServerStatus 131, 232
- ServerStatusComponent,
 - creating 378
- ServerStatusService_Proxy 366
- ServerStatusServiceAsync 367
- service implementation 363–365, 371
- service proxy 366, 372
- ServiceDefTarget 367–369
- services
 - calling from client 372–374
 - server-side
 - implementation 371–372
- <servlet> 365, 550
- servlet configuration 549–550
- servlet containers 548–549, 552
 - Apache Tomcat 548
- servlet path setting 365
- servlet tag 365
- ServletFileUpload 438
- ServletFileUpload.isMultipart-Content 438
- <servlet-mapping> 553
- servlets 326, 463
 - configuration file 549
 - mapping to URL 553
 - RPC, installing 548–553
- Set 362
- setAutoOpen() 123
- setCancelled() 428
- setCapture() 237
- setElement() 114
- setHorizontalAlignment() 161
- setHTML() 166
- setName() 427
- setOnKeepCallback() 313
- setPopupPosition() 238
- setSecondComponent() 153
- setServiceEntryPoint() 367–368
- setStyleName() 268
- setText() 166, 377
- setTextLabel() 262
- setting application's entry point 337
- setting path to public resources 325
- setting source code path 325–326
- setUrl() 128
- setVerticalAlignment() 161
- setVisibleItemCount(number) 136
- setWidget() 166, 190
- short readShort() 407
- simple panels, defined 162
- SimplePanel 164–170, 427
 - add() 153
 - insert() 153
- Singleton pattern 382
- sinkEvent() 202
- sinkEvents() 145, 147
- sinking events 197, 200–203
- sleep command 399
- Slider
 - creating 270–273
 - GridSlider 273
 - HorizontalSlider 273
 - onMouseDown() 271
 - VerticalSlider 273
- slider index 272
- Slideshow 119, 128
- software patterns 376
- source attribute 51
- SourceKeyboardEvents 252
- SourcesChangeEvent 220, 256, 272
- SourcesClickEvents 212–213, 252, 256
 - checking for instanceof 262
 - required methods 259
- SourcesEvents 214
- SourcesMouseEvent 211, 256
 - required methods 259
- SourceWriter 478, 482
- sprintf() 466
- StackPanel() 174–175
 - getSelectedIndex() 175
 - setStackText() 175

- stale connections 401
 - standard applications 566–567
 - bootstrapping 567–572
 - startUpdateTimer() 396–397
 - static-string
 - internationalization 503–516
 - status codes, checking 416
 - stdin 468
 - stopUpdateTimer() 397
 - String readString() 407
 - stringValue() 446, 460
 - strong typing 280
 - style argument 90, 96
 - adding to component 380–382
 - style DETAILED 573
 - style names, default 85
 - style PRETTY 573
 - style sheet
 - injecting 336–337
 - linking into HTML 83
 - styling 83–86
 - subclasses, serializable 358
 - submenu, reader method 419–420
 - submit button 425
 - sunk events
 - linking to
 - onBrowserEvent() 207–208
 - managing 203–207
 - super() 147
 - superclasses, serializable 358
 - Swing 25
- T**
-
- tab index 133, 168
 - Tab Listener,
 - onTabSelected() 233
 - TabBar 181
 - table performance, large table 180
 - TableListener 178, 234
 - onCellClicked() 234
 - TabListener 181
 - onBeforeTabSelect() 233
 - TabPanel 173, 181–182
 - test runners 531–532
 - TestCase
 - assertEquals() 530
 - assertFalse() 530
 - assertNotNull() 530
 - assertNull() 530
 - assertSame() 530
 - available assertions 530
 - creating new 534–537
 - fail() 530
 - failSame() 531
 - setUp() 533
 - tearDown() 533
 - writing 529–531
 - test-driven development 528
 - testing
 - asynchronous code 537–539
 - cleaning up 533–534
 - GWT code, with JUnit 528–539
 - handling errors 532–533
 - in hosted mode 86–94
 - naming methods 529
 - not meant for cleanup 533
 - running 531–532
 - setting up 533–534
 - TestRunner 531
 - text
 - capturing 432
 - capturing long 433
 - displaying as a label 131–132
 - displaying on an application 131–133
 - entering a single line 140–141
 - entering multiple lines 140
 - hiding, with PasswordTextBox 433
 - making active 132–133
 - TextArea 140, 248, 252, 433
 - setCharacterWidth() 140
 - setVisibleLines() 140
 - TextBox 140–141, 248, 252, 427, 432, 454
 - adding a KeyboardListener 262
 - cancelKey() 141
 - setMaxLength() 141
 - setVisibleLength() 141
 - TextBoxBase 115, 139, 143, 252
 - getCursorPos() 139
 - getSelectedText() 139
 - getSelectionLength() 139
 - getText() 139
 - selectAll() 139
 - setCursorPos() 139
 - setText() 139
 - theLabel.setWidth() 131
 - third-party libraries 13
 - altering module XML file 341
 - changing necessary classpaths 340
 - JavaScript, accessing 278
 - using 339–341
 - Thread.sleep() 400
 - timed events 394
 - timeout 400, 439
 - value 401
 - Timer 392, 394–395, 397
 - cancel() method 395
 - run() method 395
 - schedule() method 395
 - scheduleRepeating() method 395
 - timer events, triggering 395
 - ToggleMenuItem 151, 154–155
 - getState() 155
 - toggleShow() 204
 - tokenizer 464
 - Tomcat. *See* Apache Tomcat
 - Tooltip 231
 - _top frame name 429
 - <tr> 161
 - transient fields 358
 - TrashIcon 216
 - Tree 124–127
 - onTreeItemSelect() 236
 - setImageBase() 543
 - tree_closed.gif 542
 - tree_open.gif 542
 - tree_white.gif 542
 - TreeItem 124–127
 - getChild(int index) 126
 - getChildCount() 126
 - getChildIndex(TreeItem) 126
 - getParentItem() 127
 - getState() 126
 - getTree() 126
 - image files 542
 - images, cleaning up 542–543
 - isSelected() 127
 - onTreeItemSelected() 126
 - onTreeItemStateChanged() 126
 - setWidth 127

TreeItem(Widget)
 constructor 127
 TreeListener 126, 235
 onTreeItemStateChanges()
 235
 TreeLogger 476
 treeLogger argument 95
 tryCreate() 483
 TwoComponentMenuItem
 124, 150–153
 type structure, generating
 482–483
 typeArgs annotation
 359, 362–363
 TypeOracle 478, 486
 types
 generating 472–473

U

UIObject 115, 162
 setHeight() 116
 setPixelSize() 117
 setSize() 117
 setStyleAttribute() 116
 Uniform Resource Locator. *See*
 URL
 unit test cases 57–59
 unsinkEvents() 146–147, 205
 update() 396–397
 UpdateableComposite 395–396
 updateResultsArea() 459
 updating component 393–398
 uploading files
 FileUpload 436–439
 FormPanel 425
 URLEncoder() 469
 URI::Escape 466
 uri_escape() 467
 URL 412
 calling with
 RequestBuilder 414
 class 412
 escaping 412
 maximum length in GET 413
 query 412
 user agent 464
 User GWT module 82
 User.gwt.xml file 321

user-defined code, calling from a
 library 312–314
 UTF-8 encoding 70

V

variable scope 297
 Vector 362, 574–576
 Vector.get(int) 575
 VerticalPanel 160–161, 170,
 177–178, 253, 427

W

weak coupling 382
 web mode 573
 preparing for 95
 running an application 99
 web server, deploying to 98
 web.xml 98, 548–549
 webapps directory 550–551
 WEB-INF 549
 classes directory 551
 lib directory 551
 whitelist argument 87
 Widget class 117, 162
 WidgetCollection 170
 widgets 12–13
 adding to a page 199
 adopting 184
 applying styling 138
 basic 118
 Button 137–138, 248, 252
 ButtonBase 137
 CheckBox 138
 composite 247, 251
 creating 141–150, 197
 creating by extending an exist-
 ing widget 142, 146–150
 creating by extending the
 DOM 142–146
 definition of 111–112
 difference from panels 163
 disowning 184
 DOM elements 113–115
 FileUpload 118–119, 143–146
 FocusWidget 133–141,
 143, 167
 Frame 134
 getElement() 114

Google Video Search 306
 grouping 117
 hiding from screen 113
 hierarchy 116
 HTML 132–133, 252
 Hyperlink 119–120
 Image 127–131
 Label 131–132, 248, 252
 ListBox 135–137
 MenuBar 121–124
 MenuItem 121–124, 150–155
 NamedFrame 134
 onAttach() 199
 PasswordTextBox 140
 PNGImage 147–148, 332
 RadioButton 138–139
 removing from a page 200
 setElement() 114
 setFocus() 134
 setting style class name 113
 setting visibility of 263
 sinking events 114
 standard in GWT 115–141
 style-naming convention 268
 styling 117
 TextArea 140, 248, 252
 TextBox 140–141, 248, 252
 TextBoxBase 139, 143
 ToggleMenuItem
 151, 154–155
 Tree 124–127
 TreeItem 124–127
 TwoComponentMenuItem
 150–153
 using as Java objects 112–113
 Window 34, 290
 alert() 291
 confirm() 291
 enableScrolling() 291
 getScrollLeft() 293
 getScrollTop() 293
 open() 430
 prompt() 313
 setTitle() 128
 Window Close listeners
 onWindowClosed() 228
 onWindowClosing() 228
 WindowCloseListener
 122, 227–231

- WindowResizeListener
 - 226–227, 271
 - onWindowResizing() 227
 - windows features 430
 - writeBoolean(boolean value) 406
 - writeByte(byte value) 406
 - writeChar(char value) 406
 - writeDouble(double value) 406
 - writeFloat(float value) 406
 - writeInt(int value) 406
 - writeLong(long value) 406
 - writeObject(Object value) 406
 - writer methods 405
 - writeShort(short value) 406
 - writeString(String value) 406
- X**
-
- XML configuration file 417
 - XML data, loading with
 - RequestBuilder 417–424
 - XML handling 111
 - XMLHttpRequest 350, 410
 - XMLParser 418
- Y**
-
- Yahoo 450
 - Yahoo search API 451–453
 - parameters 451
 - Yahoo search component 453–455, 469
 - Yahoo search proxy service 461–469