

Symbols

- @ActionMapping annotation 295, 298–299
- @AfterRenderCompletion annotation 312
- @Around advice annotation 356–357
- @Aspect annotation 354–355
- @Autowired annotation 282, 288–289, 323, 330–331
- @Before annotation 327, 355
- @BeforeClass JUnit annotation 327
- @Column annotation 364–365
- @Component annotation 315
- @Configurable annotation 391
- @Constraint annotation 322
- @ContextConfiguration annotation 328
- @Controller annotation 282, 285–286, 310
- @Controller Spring Portlet MVC annotation 285
- @Controller type-level annotation 286
- @Deprecated built-in Java annotation 283
- @Deprecated Java element 283
- @Entity annotation 364–365
- @EventMapping annotation 436
- @ExceptionHandler annotation 306–307
- @GeneratedValue annotation 364
- @HandlerInterceptor annotation 312
- @Id annotation 364
- @InitBinder annotation 307–308, 310
- @InitBinder method 308, 315, 318
- @Inject annotation 288–289
- @ManagedBean annotation 520
- @ModelAttribute annotation 301, 303–305
- @NotEmpty annotation 317
- @NotEmpty constraint 321
- @ProcessAction annotation 67, 148, 499, 503
- @ProcessEvent annotation 434
- @Qualifier annotation 288–289, 323
- @RequestMapping annotation 295, 297
 - and method signatures 297
 - identifying render methods with 295–297
- @RenderMode annotation 53, 65–67, 82, 380
- @Repository annotation 286, 369
- @RequestMapping annotation 290–295
 - annotation and portlet lifecycle phases 295
 - identifying render and action methods 293–295
 - options for 290–293
- @RequestParam annotation
 - options for 300–301
 - using 299–301
- @Resource annotation 208, 282, 288–289
- @ResourceMapping annotation 455
- @RetentionPolicy 322
- @Service annotation 286
- @SessionAttributes annotation 304, 306
- @Size annotation 317
- @Size constraint 321
- @Target annotation 322
- @TestExecutionListeners 329
- @Transactional annotated methods 360
- @Transactional annotation 359–360, 371
- @Unique annotation 317, 321
- @Unique constraint 321
- @Valid annotation 314, 319–320
- @XmlRootElement class-level annotation 429
- *Util utility classes 449
 - / element 360
- \$ syntax 341
- \$/dwr/interface directory 477
- \$/contextPath/dwr/interface directory 477
- \$_javascript_folder variable 202

Numerics

-1 value 388
 2_column.png 204
 2_column.tpl file 204
 2_column.wap.tpl 204
 2-column-layouttpl folder 204

A

<a> element, HTML 508
 AbstractController class 253–256, 293
 caching portlet content 254
 rendering portlet even when minimized 255
 specifying whether PortletSession property is required 254
 synchronizing request processing 255–256
 AbstractHandlerExceptionResolver class 277
 AbstractHandlerMapping class 270–272
 default handlers 270
 handler interceptors 272
 handler mapping chains 270–272
 AbstractJUnit4SpringContextTests class 329
 Access Permission Setting subtab 560
 action attribute 55, 208, 217, 324
 action methods 83
 with @ActionMapping annotation 297–299
 with @RequestMapping annotation 293–295
 action parameters 216
 action request 54, 382
 action URLs 77–78
 ACTION_SCOPE_ID constant 95
 ActionFilter interface 495, 497, 500, 505
 ActionRequest argument 294, 332
 ActionRequest interface 92–93, 123, 127
 ActionRequest object 77–80, 504–505
 ActionRequest parameter 294
 actionRequest variable 211
 ActionRequest.ACTION_NAME parameter 78, 82
 ActionResponse interface 101, 417, 424, 428, 431
 ActionResponse object, responding to actions using 104–107
 actionResponse variable 211
 Actions button 178, 541, 546
 actions, names of 220
 action-scoped request attributes 78
 actionScopedRequestAttributes option 88, 95, 121, 130–131, 505
 scope of 125–126
 use case for 126–127
 actionStatus attribute 71, 83
 <actionURL> tag 208, 213–215, 218, 220–222, 228
 adding private render parameters to 216–220
 escaping XML in generated 220
 securing 216
 specifying
 action name 220
 portlet mode 214
 window state 214
 storing in variable 214–215
 actionURL tag 467
 ACTIVE column 337–338, 347
 Active Reverse Ajax 476
 ActiveMQ 490
 activeReverseAjaxEnabled initialization parameter 485
 Add Book form 408, 516
 Add Book hyperlink, Book Catalog portlet 438, 460, 467
 Add Book tab, Book Catalog portlet 459–461, 467, 478
 Add button 171
 Add Category panel 571
 Add Folder option 188–189
 add JavaScript and CSS to the portal page function 107
 Add New Group form 565
 Add New Portal button 559
 Add Permission button 569
 Add Portlet button 547
 Add Portlet option 547
 Add Producer button 541
 Add Producer form 542
 Add to Page check box 44
 addAnnotatedClass method 366, 368
 addBook action method 428–429
 addBook function 474, 477
 BookCatalogAjaxBean object 478–479
 MyAjaxBean object 478–479, 492
 addBook method 298, 320–321, 348–349, 413
 addBookActionUrl 324
 AddBookController class 285–288, 302–303, 326–327
 AddBookController handler 290, 296, 319, 327
 AddBookControllerTest class 326–327
 AddBookControllerTestContext class 327, 329
 addBookForm 267
 addBookForm.jsp page 296, 323
 AddBookValidator class 316–317
 addFile method 166
 Additional Files page 196
 addOnLoad function 465
 addScript method, ScriptSession object 487
 addScriptSession method 484, 487
 Administrator role 173, 175, 177–178, 192
 Administrator user 185
 afterCompletion 273
 afterRenderCompletion method 274
 afterRenderRequestComplete method 312

- Ajax (Asynchronous JavaScript and XML)
 - 437–493
 - and inter-portlet communication 489–493
 - using Comet feature 490
 - using JavaScript functions 491–493
 - basics of 439–440
 - cross-domain 488
 - portlets 440–447
 - Ajax requests 442–445
 - downloading binary content using 455–458
 - real-time 480–488
 - retrieving servlet response to update content of 445–447
 - that display date and time 442
 - resource URLs and caching 458–459
 - rich user interfaces 461–480
 - for Book Catalog portlet 459–461
 - simplified Ajax 462–480
 - securing requests 447–450
 - date/time portlet security requirements and solutions 447–448
 - portal server-specific APIs 448–450
 - support for
 - in Dojo Toolkit 465–468
 - in Portlet 2.0 450–455
- ajax function, jQuery object 469
- Ajax in portlets, benefits of using 439
- Ajax request 439
- AjaxFallbackDefaultDataTable 529
- <allow> subelement 474
- allowDuplicateParameters property 270
- annotatedClasses property 368
- annotation type parameter 323
- AnnotationConfiguration class 365–366, 368
- <annotation-driven> element 360, 371
- annotation-driven transaction management 360–361
- AnnotationMethodHandlerAdapter class 310–312
- AnnotationMethodHandlerExceptionResolver class 311–312
- annotations 285–310
 - @RequestMapping, lifecycle phases of 295
 - AspectJ, and AOP support in Spring framework 354
 - classpath scanning feature, and autoregistration 286–288
 - configuring beans, providing support for 310
 - dependency injection
 - using @Autowired and @Qualifier annotations 288–289
 - using @Resource and @Inject annotations 289–290
 - developing Book Catalog portlet using 283–284
 - handling exceptions with
 - @ExceptionHandler 307
 - identifying action methods 297–299
 - identifying controllers, with @Controller 285–286
 - identifying model attributes, using @ModelAttribute 301–304
 - creating command objects 302–304
 - reusing model attributes, using @Session-Attributes annotation 304–306
 - identifying render methods, with @RequestMapping 295–297
 - and method signatures 297
 - options for 296–297
 - mapping requests to handlers and methods, using @RequestMapping 290–295
 - annotation and portlet lifecycle phases 295
 - identifying render and action methods 293–295
 - options for 290–293
 - overview of 282–283
 - passing request parameters, using @RequestParam 299–301
 - options for 300–301
 - WebDataBinder object
 - globally initializing instances of 309–310
 - initializing using @InitBinder annotation 307–308
 - AnnotationSessionFactoryBean 368–369, 371
 - announcement feature 182
 - Announcements portlet 181–183
 - anonymous users 169, 182
 - Ant buildfile, unit testing with 325–326
 - ant deploy command 199, 202
 - Ant tool 39–41
 - JAR files 40
 - script 39–40
 - AOP (aspect-oriented programming), support in Spring framework 352–358
 - AOP framework 354
 - Apache Commons FileUpload library 157–158
 - APIs (Application Programming Interfaces) 86, 132–166
 - caching 133–141
 - expiration-based 134–137
 - validation-based 137–141
 - container-runtime options 121–131
 - actionScopedRequestAttributes 123–127
 - escapeXml 128
 - portlet and portlet application 130–131
 - renderHeaders 130
 - servletDefaultSessionScope 127–128
 - discovering portlet environment using PortletContext object 119–120
 - localizing 141–143
 - changing locales using built-in portlets 143
 - implementing portlets to support 142

- APIs (Application Programming Interfaces)
 - (*continued*)
 - portability 161–166
 - designing for 163–166
 - portal server features and limitations 162–163
 - portal server-specific 448–450
 - portlet modes 143–151
 - and markup types 150–151
 - custom 144–147
 - portal-managed and portlet-managed 148–150
 - portlet request objects 92–100
 - and lifecycle interfaces 92–94
 - handling portal server-specific requests using PortalContext object 99–100
 - PortletRequest, constants 94–97
 - PortletRequest, methods 97–98
 - portlet response objects 100–112
 - adding custom JavaScript language, and CSS files to portal page 108–112
 - choosing appropriate phase for action 107–108
 - encoding URLs, with encodeURL method 102–103
 - generating content, using RenderResponse object 103–104
 - responding to actions, using ActionResponse object 104–107
 - PortletURLGenerationListener interface 159–161
 - programmatic security 154–156
 - requirements for Book Catalog portlet 87–91
 - retrieving portlet configuration, using PortletConfig object 120–121
 - Spring Portlet MVC framework 313–316
 - storing user-specific data in portlet session 112–119
 - and content caching 118
 - obtaining session attributes, using PortletSessionUtil class 114–116
 - portlet session scopes 116–118
 - PortletSession object 112–114
 - sharing session data across portlet applications 119
 - similarities between HttpSession and PortletSession objects 118–119
 - uploading files 156–158
 - window states 151–154
 - custom 152–153
 - portlet 153–154
 - wrapper classes 158–159
 - app.server.dir property 198
 - app.server.lib.portal.dir property 198
 - app.server.portal.dir property 198
 - appAttr attribute 116
 - appendData method, ScriptBuffer object 487
 - application context
 - retrieving JNDI-bound data sources in XML files 340–342
 - application contexts
 - DI and 235–241
 - web 240–241, 244–245, 249–252
 - root 251–252
 - XML files 249–252
 - Application Programming Interfaces. *See* APIs 86
 - APPLICATION_SCOPE attribute 113, 115–117, 129, 213
 - APPLICATION_SCOPE scope, PortletSession interface 447–449
 - APPLICATION_SCOPE session attributes 413–414, 417
 - applicationClassName 531
 - applicationContext.xml 258, 370
 - applications, undeploying 46–47
 - AspectJ annotations, and AOP support in Spring framework 354
 - <aspectj-autoproxy> element 358
 - aspect-oriented programming. *See* AOP
 - associations, specifying default user 195–196
 - associative array 477
 - asynchronous Ajax request 441
 - asynchronous HTTP (Hypertext Transfer Protocol) 439
 - Asynchronous JavaScript and XML. *See* Ajax
 - asynchronous request phases 445
 - asynchronous request processing 439
 - atomicity key feature 386
 - attributes
 - model, accessing exception as 280
 - of <param> tag 223
 - AuditAspect class 354
 - AuditFields class 362, 364–365, 367
 - authentication, setting mechanism for 194–195
 - Author Name field 208
 - authorNameSearchField attribute 213
 - AUTHORS column 337
 - auto.deploy.dir property 198
 - autodetection 286, 338
 - auto-detects view 248
 - autoregistration, classpath scanning feature and 286–288
 - autowiring features 338
 - Available portlet list 543
- B**
-
- B2E (business-to-employee) applications 6–7, 376
 - backgrounds, changing color of 200–201
 - basename attribute 142

- basenames property 330
- base-package attribute 287
- BasicDataSource class 341, 343
- BasicDataSourceFactory 343
- <batchtest> element 325
- bean converter 475, 477
- beans 238
 - JSF-managed 519–520
 - PortletMultipartResolver 332–333
 - providing support for annotations
 - configuring 310–312
 - scope of defined in XML files 250–252
 - Spring framework, accessing from non-Spring objects 351–352
 - validating with JSR 303 316–323
 - configuring with Spring Portlet MVC framework 318–321
 - custom constraints 321–323
- before advice method 355–356
- before method 355
- beginTransaction method 367
- behavior, preferences 402–403
- benefits of using Ajax in portlets 439
- binary content, downloading using portlets 455–458
- BindingResult object 308, 314, 319, 321
- <body> element 521, 530
- book argument 356
- Book bean 477
- book catalog 168
- Book Catalog portlet 187–191, 335–336, 351–352, 490–491
 - database tables 336
 - BOOK_TBL table 337
 - TOC_TBL table 337–338
 - developing using annotations 283–284
 - impact of using built-in portlets 190–191
 - personalization requirements for 377–379
 - requirements for 87–91
 - rich user interfaces for 459–461
 - usage of PortletSession object in 113–114
 - using <defineObjects> tag in 212–213
 - using portlet filters with 499–500
 - using tag library to build 208–209
- Book Catalog portlet home page 379, 438
- Book Catalog web application
 - deployment descriptor 530–531
 - developing
 - using JSF 518–523
 - using Wicket portlets 528–530
 - exposing as portlet using JSF portlet bridge 523–525
 - portlet deployment descriptor 531–532
 - project structure 516–518, 526–527
 - requirements of
 - JSF 516
 - Wicket portlets 525–526
- Book Catalog-Rich Portlet 549
- book catalogs 191–192
- book category preference 401–402
- Book classes 362, 364–365, 431
- book command object 324
- Book Java object 478–479
- book JavaScript object 478–479
- book model attribute 303, 305, 319
- Book Name field 208
- Book object 348, 355–356, 361
- Book Portal administrator 176–177, 188, 191–193, 565
- Book Portal organization 170
- Book Portal page 199
- Book Portal portal 557–573
 - creating portal pages and adding portlets 568–570
 - custom portlets
 - importing into GateIn Portal server 571
 - programmatically securing 570–571
 - developing 170–178
 - creating Book Portal portal users and assigning to organization 175–177
 - mapping 174–178
 - organization for Book Portal portal 171–173
 - roles for Book Portal portal users 173–174
 - removing default home page 567–568
 - requirements for 168–169
 - setting up 558–563
 - Permission Setting tab 560–563
 - Portal Setting tab 560
 - Properties tab 560
 - users 563
 - defining default groups and memberships for registered 572–573
 - groups 563–567
- book property 520
- BOOK_ID column 337–338
- book_id column 364
- BOOK_TBL table 336–337, 361–365
- book_tbl.sql 338
- BookAddedEvent class 429–432, 434
- BookAddedEvent event 428–429, 433
- BookAddedEvent object 434
- bookAddedEvent portlet event 432
- BookAddedEventHandler 525
- bookCatalog attribute 120
- bookCatalog portlet 130–131, 150–152, 420
- bookCatalog.css 225
- BookCatalogAjaxBean object 473–480, 485, 492
- BookCatalogApp 529
- BookCatalogBean 519–520

- BookCatalogContextListener class 120, 413
 - BookCatalogPortlet class 102–104, 351–352, 570
 - BookCatalogPrefsValidator class 390–394
 - BookCatalogService class 165–166
 - bookCatalogTest-portlet.xml file 329
 - BookController 301
 - BookDao interface 346–347, 351
 - bookList.jsp page 467
 - BookPortalAdmin 561, 563–566, 569–570
 - book-portal-theme folder 199–200
 - book-portal-theme project 200–201
 - BookService beans 288–289, 326–327, 349–350
 - BookService class 120, 389, 391, 414
 - BookService interface 349–350
 - BookService methods 352, 371
 - BookServiceImpl class 349–352, 415
 - BookServlet servlet 87–88, 103, 106–107
 - bookTitle text field 227–228
 - BookUser roles 170, 173–174, 191–193
 - boolean isReadOnly method 385
 - BorderContainer layout widget 462, 464
 - breadcrumbs image, changing default 201
 - browser’s Refresh button 137
 - build directory 34, 41, 517, 526
 - build target 39, 41, 325
 - build.asarin.properties file 197
 - build.properties file 377
 - build.xml file 39
 - buildSessionFactory method 366
 - built-in Announcement portlet 183
 - built-in controllers 252–260
 - AbstractController class 254–256
 - caching portlet content 254
 - rendering portlet even when minimized 255
 - specifying whether PortletSession property is required 254
 - synchronizing request processing 255–256
 - Controller interface 253
 - ParameterizableViewController controller 256–257
 - PortletModeNameViewController controller 257–258
 - PortletWrappingController controller 259–260
 - built-in Document Library portlet 187
 - built-in handler mappings 260–272
 - AbstractHandlerMapping class 270–272
 - default handlers 270
 - handler interceptors 272
 - handler mapping chains 270–272
 - developing Book Catalog portlet 262
 - ParameterHandlerMapping implementation 264–267
 - PortletModeHandlerMapping implementation 263–264
 - PortletModeParameterHandlerMapping implementation 267–270
 - built-in Message Boards portlet 192
 - built-in portlets
 - changing locales using 143
 - impact of using 190–191
 - built-in services, reusing 162–163
 - built-in SimpleSessionStatus object 327
 - Businesses portlet 12
 - business-to-employee applications. *See* B2E
 - byId function 466
- ## C
-
- <c:forEach> tag 402
 - cache parameter, ajax function 469
 - CACHE_SCOPE property 135
 - cacheability attribute 221, 458–459
 - cacheLevelFull cacheability value 458–459
 - cacheLevelPage cacheability value 459
 - cacheLevelPortlet cacheability value 458
 - <cache-scope> element 135
 - cacheSeconds property 254
 - caching 133–141
 - and inter-portlet communication 413–415
 - content 118
 - expiration-based 134–137
 - configuring support 137
 - portlet behavior 135–136
 - portlet content 254
 - resource URLs and 458–459
 - resources 221
 - validation-based 137–141
 - Calendar built-in portlet 181
 - CannotGetJdbcConnectionException 349
 - Cascading Style Sheets. *See* CSS
 - Catalog portal page 188, 191–192, 568
 - Catalog tab, Book Catalog portlet 459, 461, 465
 - categories variable 222
 - categories, Liferay Portal 6.x 38
 - category object 386
 - category preference attribute 401–402
 - category.chapter01.helloWorld 37
 - Category-level permissions 193
 - CE (Community Edition) 539
 - center partition, BorderContainer layout widget 464
 - chaining filters 508–509
 - chains
 - handler mapping 270–272
 - handler-exception-resolver 280
 - change window title function 107
 - CharArrayWriter class 505–507
 - CharResponseWrapper class 506–508
 - class attribute 225–226, 464

- class element 205
- class files, generated 33
- classes
 - mapping to relational database tables 363–365
 - portlet, Hello World example 36–37
 - wrapper 158–159
- classpath-scanning features 285–286, 288, 338, 348, 358
- clean target 39, 325
- ClientDataRequest 93, 156
- code, setting compliance for Eclipse IDE 33
- coding step 290
- colors, of portal page background 200–201
- com.mysql.jdbc.Driver 578
- Comet feature
 - inter-portlet communication and Ajax 490
 - real-time portlets using 481–482
 - scalability problem and solution 482
 - support for Comet feature in DWR library 482–487
- CometProcessor interface 482
- command objects 285, 302–304
- commandName attribute 324
- commons-dbcp 1.3 343
- commons-pool 1.5.5 343
- CommonsPortletMultipartResolver 332
- communications
 - inter-portlet vs. inter-servlet 12–13
 - inter-portlet, and Ajax 489–493
 - See also* inter-portlet communication
- compile target 41, 325
- Compiler option 33
- complete information received 412
- compliance, setting for Eclipse IDE 33
- <component-plugin> element 572
- component-scan element 287
- concatenation string, adding parameters to portlet URLs with 223
- concurrency, issues in saving preferences 386
- conf/server.xml file 539
- ConfigurableWebBindingInitializer 310
- Configuration class 366
- Configuration option 184, 193
- Configuration permission 25
- configuring step 290
- confirmSave function 224
- connecting to database 345–346, 365–367
- Connection object 345
- <Connector> element, server.xml file 482
- consistency key feature 386
- constraints, custom 321–323
- ConstraintViolation object 320
- Consumers tab 545
- consumers, registering with WSRP producers 550–554
- <container-runtime> element 79
 - container-runtime options 78–80, 121–131
 - actionScopedRequestAttributes 123–127
 - scope of 125–126
 - use case for 126–127
 - escapeXml 128
 - portlet and portlet application 130–131
 - renderHeaders 130
 - servletDefaultSessionScope 127–128
- <container-runtime-option> 411
- containers 536
 - portlet 15–16
 - lifecycle management 15
 - portlet portability 16
 - portlet and servlet 51–52
- content
 - generation methods 64–67
 - localizing 330–331
 - preferences 400–402
- content caching 118
- content package 331
- content.messages 103, 331
- Content-disposition HTTP header 457
- ContentPane layout widget 464–465
- contextConfigLocation parameter 249–251
- ContextLoaderListener 251–252, 341, 351
- Control Panel 21, 44, 171, 175, 178, 539
- controller classes 242–243, 285–286
- Controller interface 252–253, 260, 293, 436
- controllers
 - built-in 252–260
 - AbstractController class 254–256
 - Controller interface 253
 - ParameterizableViewController controller 256–257
 - PortletModeNameViewController controller 257–258
 - PortletWrappingController controller 259–260
 - identifying with @Controller annotation 285–286
- <convert> elements 475
- converter attribute 475
- copyCurrentRenderParameters attribute 216, 218–219
- core business 5
- Create Account option 21
- create command 200, 204
- Create Database option 577
- <create> element 475
- CREATED_BY column 337
- CREATED_DATE column 337
- createdBy audit field 364
- createdDate audit field 364
- createQuery method 369
- createRenderURL method 77, 147

- createResourceURL method 453
- creator attribute 475
- cross-cutting concerns 272, 353
- CSS (Cascading Style Sheets) 437
 - JavaScript language and, including necessary in portal page 463
 - URLs for remote files 549–550
- css directory 34
- CSS elements 199
- CSS files 225
 - custom, adding to portal page 108–112
 - external 331–332
- CSS selectors 226
- Current list 543
- custom Book Portal theme 203
- custom_common.css file 201
- custom.css file 199–201
- <custom-portlet-mode> subelement 145
- <custom-window-state> subelement 153

D

- DaoFactory class 236
- DAOs (data access objects) 346–349
- data access layer 163–166
- data sources, defining 339–345
 - in application context XML file 340–342
 - JNDI-bound 342–345
- Database driver 578
- database tables
 - Book Catalog portlet 336–338
 - BOOK_TBL table 337
 - TOC_TBL table 337–338
 - Liferay Portal platform 579–581
 - permissions information in PERMISSION_ table 580–581
 - role information in ROLE_ table 579
 - user information in USER_ table 579
- database.url property 341
- databases 334–372
 - accessing Spring beans from non-Spring objects 351–352
 - Book Catalog portlet database tables 336–338
 - BOOK_TBL table 337
 - TOC_TBL table 337–338
 - combining portlets and 335–336
 - Hibernate for interaction with 361–372
 - connecting to database 365–367
 - JPA 362–363
 - mapping classes to relational database tables 363–365
 - using Spring framework with 367–372
 - JDBC module for interaction with 338–351
 - DAOs 346–349
 - database connections 345–346
 - defining data source 339–345
 - service interface and implementation classes 349–351
 - portal 577–581
 - directly using 163
 - Liferay Portal platform 577–581
 - Spring framework
 - AOP support in 352–358
 - transaction management support in 358–361
 - datasource bean 340–341, 345, 359
 - DataSource class 341, 345
 - dataSource property 345, 359
 - DataSourceTransactionManager 359–360
 - Date object, JavaScript 443
 - Date/Time portlet 440–447, 453, 549
 - DateTimePortlet class 442, 447–448, 453–454
 - DateTimeServlet servlet 443–449, 549
 - decodeAttributeName method 116
 - default action 293
 - default attribute 322
 - default handlers 270
 - DefaultAnnotationHandlerMapping class 311–312
 - defaultEmail parameter 76, 82
 - defaultErrorView name 279
 - defaultErrorView property 279
 - defaultHandler property 265, 268, 272, 312
 - <default-namespace> element 422–423, 432
 - defaultValue element 301
 - defaultViewName view 280
 - Define Permissions option 44
 - <defineObjects> tag 208–213
 - making portlet objects available with 210–212
 - using in Book Catalog portlet 212–213
 - defineObjects tag 399–400
 - Delete Node option 568
 - Delete option 547
 - DELETE request, HTTP 444
 - dependency injection
 - using @Autowired and @Qualifier annotations 288–289
 - using @Resource and @Inject annotations 289–290
 - dependency injection. *See* DI
 - DependencyInjectionTestExecutionListener 329
 - depends attribute 39
 - deploy directory 18
 - deployment descriptors 72–73
 - of portlets 243–244
 - configuring PreferencesValidator interface in 392–395
 - defining events in 431–432
 - defining multiple portlets in 410–411
 - JSF 523–525
 - setting public render parameters in 419–423
 - Wicket 531–532

- deployment descriptors (*continued*)
 - of web applications 245–246
 - Wicket web application 530–531
 - <description> subelement 97
 - designing, for portability 163–166
 - data access layer 163–166
 - request-handling logic 163
 - service layer 163
 - destroy method 59, 496–497, 499
 - destroy-method attribute 341
 - detectPortletContext 530–531
 - development environments 26–35
 - configuring Eclipse IDE 26–33
 - creating new server profile 26
 - for JDK 5.0 or later 31–33
 - server 28–30
 - setting source and generated class file compliance 33
 - specifying installation directory of Tomcat server 27–28
 - project structure 34–35
 - DI (dependency injection), and application context 235–241
 - e-commerce example 235–237
 - web application contexts 240–241
 - diffs directory 199–202
 - Dijit library 462
 - Direct Web Remoting. *See* DWR
 - directories, specifying for Tomcat server 27–28
 - DirtyContextTestExecutionListener 329
 - discussion forums, settings for displaying 192–193
 - dispatch events to portlets function 107
 - dispatcher.include 106
 - DispatcherPortlet class 246–252, 276–277, 331–332
 - Display Date option 183
 - Display Order option 180
 - Distribution Scope drop-down list 182–183
 - div element, HTML 443
 - <div> tag 444, 447, 471
 - div tag 226, 464
 - DLLocalServiceUtil class 166
 - Document Library portlet 188–191
 - document object 447
 - Document Object Model. *See* DOM
 - doDispatch method 65
 - doEdit method 61, 65
 - doFilter method 496–498, 505, 509
 - doGet method 114, 444
 - doHeaders method 99, 111, 331–332
 - doHelp method 61, 65
 - Dojo parser 465
 - Dojo Toolkit, simplified Ajax with 462–468
 - adding widgets to JSP page 463–465
 - Ajax support in Dojo Toolkit 465–468
 - including necessary JavaScript language and CSS 463
 - Dojo, iFrame proxy 488
 - dojo.js file 463
 - dojoType attribute 464–465
 - Dojox 462
 - dojox directory 463
 - DOM (Document Object Model) 437
 - doSomething method 294–295, 299, 314
 - dot (.) Java bean 370
 - doView method 61, 65, 148
 - download files function 108
 - Download link, Book Catalog portlet 460
 - downloading binary content, using portlets 455–458
 - doXXXX methods 63, 66
 - driverClass property 341
 - DWR (Direct Web Remoting) 451, 462, 474
 - integration options for 488
 - simplified Ajax with 471–480
 - configuring DWR library 473–476
 - including necessary JavaScript files 476–477
 - support for Comet feature in 482–487
 - <dwr> element 474
 - dwr.util.getValue function 478
 - dwr.xml file 474, 477, 479–480
 - DwrServlet object 475–476, 479, 484–485
 - dynamic content 438
- ## E
-
- Eclipse IDE
 - configuring 26–33
 - creating new server profile 26
 - for JDK 5.0 or later 31–33
 - server 28–30
 - setting source and generated class file compliance 33
 - specifying installation directory of Tomcat server 27–28
 - project structure 34–35
 - Eclipse projects 33
 - e-commerce, example of DI and 235–237
 - Edit Layout option 558
 - EDIT mode 258, 379–380, 399
 - Edit Navigation option 558
 - Edit option 544, 547
 - Edit Permission Setting tab 561–562
 - Edit Portal's Config option 558
 - EDIT portlet mode 62–63, 187, 292, 502
 - Edit Registration option 547, 552
 - editBook action method 305–306
 - editBookController 263, 275–276
 - EditBookController class 301, 315
 - EditBookController handler 299, 304–305

EE (Enterprise Edition) 539
 encodeURL method 102–103
 encoding URLs with 102–103
 PortletResponse object 456
 encoding, URLs 102–103
 engine.js file 476–477, 483–484
 Enterprise Edition. *See* EE
 entry elements 257
 error property 467
 errorMap property 389, 413, 522
 errorMessage.replace 397
 Errors object 307
 <errors> tag 324
 escapeXML attribute 220
 escapeXml container-runtime option 122, 128
 ETAG field 93
 Event object 434
 EventAwareController 260, 436
 event-based inter-portlet communication 424
 <event-definition> element 432–433
 EventFilter interface 495
 EventPortlet interface 92, 432–435
 eventRequest variable 93, 211, 434
 EventResponse interface 101, 428, 434–435
 EventResponse objects 216, 423–424, 431, 434
 eventResponse variable 211
 events
 defining in portlet deployment descriptor
 431–432
 setting for publishing 428–430
 exception class, mapping to view based on
 278–280
 accessing exception as model attribute 280
 exception mappings 278–280
 handler mappings 279
 handler-exception-resolver chain 280
 exception mappings 278–280
 exceptionAttribute property 280
 exceptionMappings property 278
 exceptionMsg render parameter 107
 ExceptionResolver 332
 exceptions, handling 276–280
 during request handling 277–278
 mapping to view based on exception class
 278–280
 with @ExceptionHandler annotation 307
 exclude-filter subelement 287
 ExoContainer object 570
 Expiration Date option 183
 expiration-based caching 134–137
 configuring support 137
 portlet behavior 135–136
 <expiration-cache> 410, 425
 Export/Import subtab 179

eXtensible Hypertext Markup Language files. *See*
 XHTML
 Extensible Markup Language files. *See* XML
 EXTENSION column 338
 ExternalContext's redirect method 520, 523

F

<f:ajax> tag 522
 faces-config.xml file 518
 FacesServlet 518
 fadeIn function 471
 fadeOut function 471
 failedKeys 392
 <field> elements 572
 fields, hidden form 267
 file JavaScript object 480
 FILE_NAME column 337
 FileItem object 157
 FileItemStream object 158
 fileName property 480
 files
 generated class, setting compliance for Eclipse
 IDE 33
 remote, CSS and JavaScript 549–550
 uploading 156–158
 Apache Commons FileUpload library
 157–158
 PortletMultipartResolver bean 332–333
 XHTML 520–523
 fileset element 39
 FileUpload library 156–158, 332
 filter chaining 508–509
 <filter> element 502
 filterActionURL method 160
 FilterChain object 497–499, 501, 505
 <filter-class> element 502
 FilterConfig object 497, 501, 503
 <filter-mapping> element 503, 508, 531
 filterMappingUrlPattern parameter 531
 <filter-name> element 502–503
 filterRenderURL method 160
 filterResourceURL method 160
 filters, portlet. *See* portlet filters
 firstName field 85
 forEach JSTL tag 215
 form field values, obtaining in portlet class
 227–228
 GlassFish server 228
 Jetspeed portal 228
 Liferay Portal portal 227–228
 form fields, hidden 267
 <form> tag 84, 324
 form tag library 323–324

forms, validating 313–323
 beans with JSR 303 316–323
 Spring Validation API 313–316
 Forums portal page 193
 forward method 74
 FreeMarkerViewResolver 247

G

gadgets, defined 13
 GateIn Portal server 555–573
 Book Portal web portal 557–573
 creating portal pages and adding
 portlets 568–570
 custom portlets 570–571
 removing default home page 567–568
 setting up 558–563
 users 563, 572–573
 installing 555–556
 user interface 556–557
 <gatein-install-dir>directory 572
 generate markup content function 107
 generateUrls method 147, 153
 GenerationType.AUTO 364
 GenericFacesPortlet 524
 GenericPortlet class 36, 60–62, 64–67, 331–332
 GET method, HTTP 499, 503
 GET request, HTTP 443–444, 466, 470
 getAttribute method 119, 159
 getAttributeNames() method 116
 getBooks function 465–467
 getBooks methods 347, 349, 352, 370, 401, 520
 getBooksByCategories method 401
 getCommandObject method 303
 getContainerRuntimeOptions method 120–121
 getCurrentSession method 357, 370–371
 getElementId method 447
 getETag method 139
 getEvent method 434
 getFailedKeys method 397
 getHomePage method 528–529
 getInitParameter method 76, 119, 121, 501
 getInitParameter(String name) method 121
 getInstance method 484, 486–487
 getItemIterator method 158
 getItemIterator(ActionRequest) method 158
 getLocale method 75, 142, 331
 getMessage method 320
 getMimeType method 120
 getNamedDispatcher method 120
 getNamedDispatcher(String name) 74
 getNamespace method 227
 getObserverPortlet method
 BookCatalogAjaxBean object 493
 MyAjaxBean object 492
 getParameter method 217, 219
 getParameter(String name) 423
 getParameterMap method 98
 getParameterValue method 382
 getParameterValues method 222, 382
 getParameterValues(String name) 423
 getPortalContext method 98
 getPortalInfo method 99, 111, 163
 getPortletConfig method 75, 142
 getPortletContext method 121
 getPortletMode method 64, 98, 263
 getPortletName method 121
 getPortletOutputStream method 454
 getPortletSession method 98
 getPortletSession(false) 254
 getPreferences method 98, 383, 398
 getPrivateParameterMap method 98
 getProcessingEventQNames method 433
 getProperties method 98
 getProperty method 98, 229
 getPropertyPath method 320
 getPublicParameterMap method 98, 423
 getPublicRenderParameterNames method 423
 getPublishingEventQNames method 432
 getReader methods 158
 getRecentBook JavaScript function 491, 493
 getRecentBook method 414–415
 getRemoteUser method 154, 449
 getRequest method 159
 getRequestDispatcher(String path) 74
 getResourceBundle method 104, 141
 getResourceBundle(java.util.Locale) method 75
 getResourceID method 457
 getSessionFactory method 366–367
 getString methods 142
 getSupportedLocales method 121
 getSupportedPortletModes method 100, 145, 148
 getSupportedWindowStates method 100, 152
 getTarget method 357
 getUserPrincipal method 154
 getUserRoles(long userId) method 449
 getValue method 384, 402, 434
 getWebApplicationContext method 352
 getWindowID method 115
 getWindowState method 98
 getWriter method 36, 505
 RenderResponse object 506
 RenderResponseWrapper object 507
 GlassFish project, with OpenPortal portlet
 container 430
 GlassFish server 228
 GlassFish Server v3.0.1 portal server 87, 109, 124
 global attribute 344
 <GlobalNamingResources> element 343
 Gmail portlet 376–377

Gmail web application 377
 greetingMessage, preference 400, 402–403
 Greetings Message personalization option 378
 Group ID 172
 Group Management tab 565
 groups
 and memberships, defining default for registered users 572–573
 associating users to 566–567
 custom 563–566
 Guest role 170, 187, 192–193
 Guest users 186, 191–192

H

<h:body> tag 523
 <h:head> tag 523
 <h:inputText> tag 522
 <h:outputScript> tag 521, 523
 <h:outputStylesheet> tag 521, 523
 handleActionRequest method 242, 246
 handleActionRequestInternal method 254–256
 handleAs property 466
 handleEventRequest method 436
 handler interceptors 260
 handler mappings 279
 handleRenderRequest method 242–243, 245–246, 257
 handleRenderRequestInternal method 254–255
 HandlerExceptionResolver bean 310
 handler-exception-resolver chain 280
 HandlerExceptionResolver interface 276–277
 HandlerExecutionChain 261, 277
 HandlerInterceptors 260, 273
 HandlerMapping beans 246, 248, 281
 HandlerMapping interface 246–247, 270, 275
 handlers
 built-in mappings 260–272
 AbstractHandlerMapping class 270–272
 developing Book Catalog portlet 262
 ParameterHandlerMapping
 implementation 264–267
 PortletModeHandlerMapping
 implementation 263–264
 PortletModeParameterHandlerMapping
 implementation 267–270
 default 270
 interceptors 272–276
 implementing and configuring 272–273
 processing requests using 273–276
 mapping chains 270–272
 mapping requests, to using @RequestMapping
 annotation 290–295
 annotation and portlet lifecycle phases 295
 identifying render and action methods
 293–295
 options for 290–293
 hasErrors method 314
 HashMap 413
 hasPermission 570
 HEAD element 108
 <head> element 109, 521, 530
 HEAD request, HTTP 444
 <HEAD> tag 109
 HeidiSQL client, MySQL database and
 installing 574–577
 Hello World examples, portlets 35–47
 adding to portal page 43–46
 Liferay Portal 6.x 38–39
 portlet class 36–37
 portlet settings 37
 resource bundle 37–38
 with Ant tool 39–41
 with Maven tool 42–43
 Hello World message 243
 Hello World portlet 241–246
 controller class 242–243
 JSP page 243
 portlet deployment descriptor 243–244
 web application context 244–245
 web application deployment descriptor
 245–246
 helloWorld portlet 249
 helloWorld view name 245
 HelloWorldController class 242, 245–246, 252
 HelloWorldController portlet 242
 helloWorldMessage request attribute 243
 HelloWorldPortlet class 36, 40
 HELP portlet mode 62–63, 502
 help.jsp 70
 Hibernate framework 361–362
 Hibernate ORM framework, interaction with
 databases 361–372
 connecting to database 365–367
 JPA 362–363
 mapping classes to relational database
 tables 363–365
 using Spring framework with 367–372
 Hibernate Query Language. *See* HQL
 Hibernate Session 366–367
 HibernateBeanConverter 488
 HibernateBookDao class 368–370
 HibernateTocDao 370
 HibernateTransactionManager 359–360, 371
 hidden form fields 267
 high latency 481
 Home class 528
 Home hyperlink 49

home pages 23, 262
 Book Portal 180, 183
 removing default 567–568
 Home portal page 178–181
 home.jsp page 449, 454, 464, 478–479, 484
 HQL (Hibernate Query Language) 370–372
 href attribute 76
 HSQldb database 19, 336, 574
 HTML (Hypertext Markup Language) 439, 444, 528
 HTML code 214
 HTML combo box 381
 <html> element 523, 530
 HTML elements 226
 html method 469
 html-portlet 526
 html-web 526
 HTTP (Hypertext Transfer Protocol) 442–443, 480–481
 HTTP protocol 216, 285
 HTTP request parameters 285
 HTTPS (Hypertext Transfer Protocol Secure) 439
 HttpServletRequest interface 98
 HttpServletRequest object 210, 479
 HttpServletResponse object 444, 450
 HttpSession class 112–113, 128
 HttpSession object 118–119, 210, 450
 HyperlinkFilter class 507–508
 hyperlinks, converting text to with portlet filters 505–508
 Hypertext Markup Language. *See* HTML
 Hypertext Transfer Protocol Secure. *See* HTTPS
 Hypertext Transfer Protocol. *See* HTTP

I

<i> tag 444
 IBM's WebSphere Portal 16
 id attribute 238, 364
 id element 205
 <identifier> element 420–421
 IDEs (integrated development environments)
 configuring 26–33
 project structure 34–35
 <iframe> HTML element 513
 iFrame portlets 513–515
 iFrame proxy 488
 <iframe> tag 109
 iFrame-based portlet 513
 IllegalArgumentException 278
 IllegalMonitorStateException 278
 IllegalPathStateException 278
 IllegalStateException 278, 307
 IllegalThreadStateException 278
 images directory 34
 images, breadcrumbs 201
 Import Applications button 571
 Inband option 553
 in-band registration, examples of 551–553
 <include> elements 475
 include method 74
 include-filter subelement 287
 In-Demand Book Apps web application 169
 init method 58, 60, 81, 352, 496, 499
 initialization parameters 76
 initialize method 323
 <init-param> element 73, 76, 503
 innerHTML property 447, 466, 471
 <input> tag 324
 InputStream type 480
 Install More Portlets button 540
 installation directories, of Tomcat server 27–28
 installing, Liferay Portal 6.x 18–20
 <instanceable> element 38, 60
 int Java type 367
 integrated development environments. *See* IDEs
 integration, options for DWR library 488
 interceptors property 272
 interceptors, handler 272–276
 implementing and configuring 272–273
 processing requests using 273–276
 InternalResourceViewResolver bean 245, 247–248, 252
 International Standard Book Number. *See* ISBN
 internationalization, using resource bundles 75
 inter-portlet, communication 295, 405, 407, 415–416, 426–436
 and Ajax 489–493
 using Comet feature 490
 using JavaScript functions 491–493
 example 407–409
 need for 406–407
 portlet events 425–432
 advantages and disadvantages of 435–436
 receiving 432–435
 sending 427–432
 support in Spring Portlet MVC interface 436
 PortletSession interface 409–417
 advantages and disadvantages of 416–417
 defining multiple portlets 410–411
 example 415–416
 storing and retrieving information from 411–415
 public render parameters 417–425
 advantages and disadvantages of 424–425
 example 425
 methods related to 423–424
 setting and retrieving 418–423
 inter-portlet, communication scenarios 411
 Inventory portlet 406

- IoC (Inversion of Control) 237
 - ISBN (International Standard Book Number) 460, 480
 - ISBN fields 284
 - ISBN value 284
 - ISBN_NUMBER column 337
 - isbnNumber parameter 215, 219, 222
 - isbnNumber property 321
 - isbnNumber request parameter 300
 - isFormField method 158
 - isMarkupValid method 138–139
 - isMultipartContent(ActionRequest) method 157
 - isPortletModeAllowed method 98, 145–147
 - isUniqueISBN method 349, 391
 - isUserInRole method 154, 156, 454
 - isValid method 323
 - isWindowStateAllowed method 98, 153
- J**
-
- JAR (Java Archive) files 40
 - Java class 354
 - Java Database Connectivity. *See* JDBC
 - Java Development Kit. *See* JDK
 - Java EE container 481
 - Java EE environment 363
 - Java Naming and Directory Interface. *See* JNDI
 - Java NIO API 482
 - Java Persistence API. *See* JPA
 - Java portlet technology 376–377
 - Java SE 5 features 33
 - Java Specification Request. *See* JSR
 - Java Transaction API. *See* JTA
 - JAVA_HOME environment variable 19, 42
 - java.io.CharArrayWriter object 506
 - java.io.Serializable 428–429
 - java.lang.Long type 300
 - java.nio package 482
 - java.sql.SQLException 349
 - java.sql.Statement object 349
 - java.util.Map 165, 210, 389
 - java.util.Map objects 248, 257, 423, 479
 - java.util.Properties 264
 - java.util.ResourceBundle class 103
 - java.util.ResourceBundle object 142
 - JavaBean-style attribute 364
 - JavaScript 437–438
 - javascript attribute 475
 - JavaScript files
 - adding as part of portal theme 202
 - external 331–332
 - URLs for remote 549–550
 - JavaScript functions, inter-portlet communication 491–493
 - JavaScript language
 - and CSS 463
 - custom 108–112
 - files 476–477
 - JavaScript libraries 461–462
 - JavaServer Faces. *See* JSF
 - JavaServer Pages. *See* JSPs
 - javax.naming.InitialContext object 342
 - javax.naming.spi.ObjectFactory interface 343
 - javax.portlet.action parameter 499–500, 503, 505
 - javax.portlet.Action request parameter 220
 - javax.portlet.action request parameter 298–299
 - javax.portlet.actionScopedRequestAttributes 78–80, 95, 123, 389
 - javax.portlet.as render parameter 95
 - javax.portlet.escapeXml container-runtime option 129
 - javax.portlet.EventPortlet interface 57, 436
 - javax.portlet.faces.defaultViewId.edit 524
 - javax.portlet.faces.defaultViewId.<portletMode> initialization parameter 524
 - javax.portlet.faces.defaultViewId.view initialization parameter 524
 - javax.portlet.GenericPortlet class 243
 - javax.portlet.p.XX?myAttr format 115
 - javax.portlet.Portlet interface 35, 57
 - javax.portlet.PortletConfig object 58
 - javax.portlet.RenderResponse object 36
 - javax.portlet.ResourceServingPortlet interface 57
 - javax.portlet.title 37, 141, 244
 - javax.sql.DataSource bean 339, 341, 345
 - javax.validation.Validator interface 318, 320
 - javax.xml.namespace.QName 428
 - JDBC (Java Database Connectivity) module
 - 338–351
 - DAOs 346–349
 - database connections 345–346
 - defining data source 339–345
 - service interface and implementation classes 349, 351
 - jdbc.default.driverClassName 578
 - jdbc.default.password 578–579
 - jdbc.default.url 578
 - jdbc.default.username 578–579
 - jdbc/myportalDB data source 344
 - JdbcBookDao class 347, 349–350, 361
 - JdbcTemplate class 338, 345–347, 349, 368
 - JdbcTocDao 349–350
 - JDK (Java Development Kit) 5.0 31–33
 - jee schema 344–345
 - Jetspeed 2.2 portal server 109, 122, 124
 - Jetspeed portal 228
 - Jetspeed-2.2.1 portal server 87
 - JetspeedPortalService 164
 - Jetty 6 482, 488

- JNDI (Java Naming and Directory Interface)
 - 342–345
 - <jndi-lookup> element 342, 344–345, 359
 - jndi-name attribute 344
 - JPA (Java Persistence API) 362–363
 - JpaTransactionManager 360
 - jQuery library 468–471
 - jQuery object 469
 - jQuery selector 469
 - jquery.js file 468
 - JRE Definition wizard page 31
 - JRE option 28
 - JRE Type page 31
 - js directory 34
 - JSF (JavaServer Faces), Book Catalog web
 - application 515–525
 - developing using JSF 518–523
 - exposing as portlet using JSF portlet
 - bridge 523–525
 - project structure 516–518
 - requirements of 516
 - JSONP (JSON with Padding) 488
 - JSPs (JavaServer Pages) 243
 - adding widgets to 463–465
 - creating to display portlet content 70–71
 - obtaining preferences in 397–400
 - pages 84–85
 - JSR (Java Specification Request) 303 316–323
 - configuring with Spring Portlet MVC
 - framework 318–321
 - custom constraints 321–323
 - JTA (Java Transaction API) 359
 - JTA transaction manager 371
 - JtaTransactionManager 361
 - <junit> element 325
- L**
-
- <language> element 142
 - Language-ext_pt_BR.properties file 38
 - Language-ext.properties file 35
 - layout templates 203
 - layouts, of pages 203–206
 - layoutpl folder 197
 - leasing, WSRP registrations 553–554
 - Let's Go Home link, Book Catalog portlet 460
 - lib directory 34
 - libraries, form tag 323–324
 - library attribute 521
 - lifecycle 48–85
 - of portlet filters 497–499
 - portlets
 - development 68–80
 - generating content based on portlet
 - mode 62–68
 - GenericPortlet class 60–62
 - Portlet interface 57–60
 - URLs 54–57
 - vs. servlets 51–54
 - User Registration portlet 49–50, 80
 - <lifecycle> element 503
 - lifecycle interfaces, portlet request objects
 - and 92–94
 - LIFECYCLE_PHASE constant 94–95
 - Liferay Plugins SDK 196, 203
 - developing portals with 196–206
 - configuring Plugins SDK 197–198
 - page layout templates 203, 206
 - portal themes 198–203
 - Liferay Portal 18–26, 447, 449, 538, 541, 543, 546
 - adding portlets to pages 23–24
 - categories 38
 - configuring server settings for 193–196
 - setting authentication mechanism 194–195
 - specifying default user associations 195–196
 - creating public portal page in 21–23
 - features specific to 38–39
 - installing 18–20
 - registering users with 20
 - roles and permissions in 24–26
 - Liferay Portal installation 170, 173, 178, 200–202, 448
 - Liferay Portal platform 538–539, 574–581
 - installing MySQL database and HeidiSQL
 - client 574–577
 - portal databases 577–581
 - Liferay Portal portal 227–228
 - Liferay Portal server 193–194, 196, 204
 - Liferay Portal themes 199
 - Liferay Portal wiki 195
 - LIFERAY_HOME/tomcat-6.0.26/bin directory 19
 - liferay.portal.home property 40–41, 49, 515, 526
 - liferay-display.xml file 35
 - liferay-plugin-package.properties file 200
 - LiferayPortalService 164, 166
 - liferay-portlet.xml file 35, 448, 463, 476
 - liferay-portlet-app element 38
 - Liferay-Tomcat 539
 - Lifetime Supplied option 554
 - <link> element 108–109
 - <link> tag 550
 - list method 369
 - <listener> element 161
 - listeners, URL generation 159–161
 - load property 466
 - localhost 556, 575, 578
 - localization, using resource bundles 75
 - localizing 141–143
 - changing locales using built-in portlets 143
 - content 330–331
 - implementing portlets to support 142

locally registering, remote portlets 547–550
 LocalValidatorFactoryBean class 318–321, 329
 Location portlet 12
 locations attribute 328
 locations, changing using built-in portlets 143
 logged-in user 377
 Logger method 357
 LoggingAspect aspect 356
 loggingInterceptor 273
 logic, request-handling 163
 logMessage method 356–357
 Long Java type 367
 Long type 309
 long-lived connections, Comet 481
 LongNumberPropertyEditor property 309–310
 Look and Feel subtab 179
 lowest order value 270

M

main method 366
 Make It Public (Everyone Can Access) check box 569
 Manage Entries tab 182
 Manage Pages option 178
 Manage Portlets option 547
 Manage Portlets screen 547
 managed beans 518–520
 manager membership 562, 565
 manager role 561
 mappedHandlerClasses properties 279
 mappedHandlers property 279
 mappings
 built-in handler 260–272
 AbstractHandlerMapping class 270–272
 developing Book Catalog portlet 262
 ParameterHandlerMapping
 implementation 264–267
 PortletModeHandlerMapping
 implementation 263–264
 PortletModeParameterHandlerMapping
 implementation 267–270
 chains handler 270–272
 classes to relational database tables 363–365
 requests to handlers and methods using
 @RequestMapping annotation 290–295
 annotation and portlet lifecycle phases 295
 identifying render and action methods
 293–295
 options for 290–293
 roles to Book Portal portal organization
 174–175
 to view based on exception class 278–280
 accessing exception as model attribute 280
 exception mappings 278–280
 handler mappings 279
 handler-exception-resolver chain 280
 mappings property 263
 mapRow method 348
 Markup interface 537
 markup types 150–151, 153–154
 MARKUP_HEAD_ELEMENT 108–111, 550
 mashup portlet 407
 mashups 15
 match attribute 475
 Maven tool 42–43
 maxActive attribute 343
 maxIdle attribute 343
 Maximum Number of Books to Show
 preference 400
 maxNumOfBooks preference 403
 maxWaitAfterWrite initialization parameter 485
 memberships, groups and 572–573
 Message Boards portlet 192–193
 Message class 477
 Message object 479
 <message> tag 75
 MessageInterpolator 322
 MessageSource bean 143
 messageSource bean 330–331
 messageSource interface 330
 messageSource property 330–331
 MessageSourceAware interface 330–331
 messageText element 447
 <meta> element 109
 metadata-driven approach 361
 method attribute 54
 method level 285
 method parameter 285
 method signatures 297
 <method_name> 65
 method-level @RequestMapping annotation
 292–293
 methods
 content generation 64–67
 mapping requests using @RequestMapping
 annotation 290–295
 annotation and portlet lifecycle phases 295
 identifying render and action methods
 293–295
 options for 290–293
 related to public render parameters 423–424
 MimeResponse interface 101, 451
 MimeResponse.EXPIRATION_CACHE
 property 254
 <mime-type> element 151
 mock objects, unit testing with 326–327
 MockActionResponse object 327
 MockBookService class 327
 MockRenderResponse instance 327

MockRenderResponse object 327
 model attributes
 accessing exception as 280
 identifying using @ModelAttribute
 annotation 301–304
 creating command objects 302–304
 reusing model attributes 304–306
 Model object 301
 ModelAndView constructor 243
 ModelAndView object 242–243, 245–247, 276–277
 modes, portlet 143–151
 and markup types 150–151
 custom modes 144–147
 defining support for 67–68
 generating content based on 62–68
 portal-managed and portlet-managed
 modes 148–150
 setting 145–147
 specifying 214
 MODIFIED_BY column 337
 MODIFIED_DATE column 337
 modifiedBy audit field 364
 modifiedDate audit field 364
 MultipartActionRequest 332
 mvn clean install command 42
 My Consumer WSRP consumer 547
 My Hello World portlet 45
 My Home Page portal page 24
 My Pages option 21
 My Producer provider 544
 My Producer WSRP producer 545
 my_breadcrumbs.png file 201
 MyAbstractHelloWorldController class 256
 myaction attribute 88, 104, 208–209, 212–213
 myaction parameter 265–269, 271, 275
 myaction private render parameter 418
 myaction request parameter 88, 296, 298, 500–501
 MyActionRequestWrapper class 159
 MyAjaxBean object 475–476, 478–480, 492
 MyAjaxBean.js file 476–477
 myAttr attribute 116
 myBookISBN variable 420–422
 myBookService id 289
 myCustomRecentBookIsbnParamName render
 parameter 421–422
 MyDispatcherPortlet class 332
 myExceptionHandler view name 278
 MyHandlerInterceptor class 312
 my-home-page 25
 myparam request parameter 159
 myportaldb database 338, 359, 577–579
 myPortletUrl portlet 267
 MyPropertyEditorRegistrar 310
 MyRenderRequestWrapper class 159
 MySamplePortal 171–172, 178, 180, 203

MySQL database 574–575, 577, 579
 and HeidiSQL client, installing 574–577
 configuring Liferay Portal platform with
 577–579
 MyUniqueConstraintValidator class 322–323
 MyUnknownStateException 278
 MyUrlGenerationListener class 160–161
 MyValidator class 390
 myValidator class 315

N

name attribute 223, 344
 <name> element 76, 97, 404, 432
 name element 364
 Name text field 543
 NamedParameterJdbcTemplate class 346
 <namespace> tag 223–228, 444
 GlassFish server 228
 Jetspeed portal 228
 Liferay Portal portal 227–228
 portlet class 227–228
 namespaces, values of 228
 namespace-uri 434
 Navigation Management window 567
 .net value 402
 New Server dialog box 26
 new.user.event.listener 572
 no information received 412
 no-argument constructor 364
 numberOfCachedScopes 125, 127

O

object type parameter 323
 object-oriented terms 353
 onreadystatechange event listener 445–446
 open method 443, 446
 OpenPortal Portlet Container 538
 OpenPortal WSRP consumer creation screen 553
 OpenPortal WSRP producer portlet 552
 openSession method 367, 370
 OpenSSO user interface 194–195
 openStream method 158
 optional attribute 216
 optional value element 286
 OPTIONS request, HTTP 444
 Order portlet 406
 order property 271, 280
 OrderDAO application object 235–236, 240
 OrderDAOIntf attribute 236, 238
 OrderDAOIntf interface 236, 240
 orderDAOIntf property 240
 OrderHandler application object 235

OrderHandler class 236–237, 239
 OrderService class 236–238, 240
 OrderService object 235, 239–240
 org.apache.commons.dbcp.BasicDataSource 341
 org.apache.coyote.http11.Http11NioProtocol
 protocol 482
 org.portletfaces.bridge.BridgeEventHandler
 interface 525
 org.portletfaces.bridge.GenericFacesPortlet 524
 org.springframework.validation.Validator 315,
 318–319
 org.springframework.web.portlet.Dispatcher-
 Portlet class 243
 org.springframework.web.portlet.mvc.Controller
 interface 242, 436
 organization
 associating custom role with 174–175
 creating Book Portal users and assigning
 to 175–177
 for Book Portal portal 171–173
 mapping roles to Book Portal portal 174–175
 Organizations option 171, 176, 178
 Organizations window 171
 Outband options 553
 out-of-band registration, examples of 551–553
 <outputScript> tag 518
 <outputStylesheet> tag 518
 overriding database 578

P

P3PUserInfos nested class 97
 Page Creation wizard 568
 Page Layout option 181, 205
 page layout templates 203–206
 Pages subtab 179
 pages, in Liferay Portal 6.x
 adding portlets to 23–24
 creating public portal pages 21–23
 <param> element 475
 <param> tags 208, 215–216, 221–223, 267
 ParameterHandlerMapping 264–267, 273
 ParameterizableViewController controller
 256–257
 parameterizedView.jsp JSP page 257
 parameterMap properties 266
 ParameterMappingInterceptor 273, 275, 312
 parameterName property 265–266, 275
 parameters 79, 223
 params elements 291, 299
 parse function 465
 password property 341
 path attribute 324
 PATH environment variable 42
 Permission Setting tab 560
 PERMISSION_ table 580–581
 permissions
 in Liferay Portal 6.x 24–26
 setting for portal pages 185–187
 for displaying book catalog 191–192
 for displaying discussion forums 192–193
 setting for portlets
 for displaying book catalog 191–192
 for displaying discussion forums 192–193
 permissions information, in PERMISSION_
 table 580–581
 Permissions tab 187
 Permissions table 25
 persistent stores, saving preferences to 383–384
 personalization 375–404
 introduction to 376–377
 options in EDIT mode 379–381
 adding support for 379–380
 writing render method for 380–381
 preferences 382–384
 behavior 402–403
 content 400–402
 defining in portlet.xml file 403–404
 retrieving 382–383, 397–404
 saving 383–388
 validating 388–397
 requirements for Book Catalog portlet 377–379
 phases, choosing appropriate for action 107–108
 plain text response 444
 pluggable user interface components 11
 Plugins SDK directory structure 197–198
 plus sign (+) button 565
 pointcut expression 353
 POJO-based development 235
 polling, real-time portlets using 481
 Popular Java Books 185
 portability 161–166
 designing for 163, 166
 data access layer 163–166
 request-handling logic 163
 service layer 163
 of portlets 16
 portal server features and limitations 162–163
 directly using portal databases 163
 reusing built-in services 162–163
 portal administrator 168–169, 191
 portal databases 577–581
 configuring Liferay Portal platform with MySQL
 database 577–579
 Liferay Portal platform database tables 579–581
 permissions information in PERMISSION_
 table 580–581
 role information in ROLE_ table 579
 user information in USER_ table 579

- portal pages 178–193
 - adding custom JavaScript language and CSS files to 108–112
 - creating and adding portlets to 568–570
 - Home 178–181
 - including necessary JavaScript language 463
 - setting permissions for 185–187
 - book catalog 191–192
 - discussion forums 192–193
- portal server 11, 13, 536
- portal server-specific 111–112, 198, 409
- Portal Setting tab 560
- portal_normal.vm file 202
- portal.admin 563–564, 566–567, 571
- PortalAdministrator role 170–171, 173
- PortalContext object 98–99, 108
- portal-ext.properties file 417, 448
- <portal-managed> element 149
- portal-managed portlet 148–150, 187
- portals 167–206
 - benefits of 6–10
 - enriched user experience 6–9
 - unified information view 9–10
 - Book Portal
 - developing 170–178
 - requirements for 168–169
 - changing background color of portal page 200–201
 - configuring Liferay Portal server settings 193–196
 - setting authentication mechanism 194–195
 - specifying default user associations 195–196
 - description of 4–6
 - developing with Liferay Plugins SDK 196–206
 - configuring Plugins SDK 197–198
 - page layout templates 203–206
 - portal themes 198–203
 - Liferay Portal 6.x 38–39
 - pages
 - adding portlets to 43–46
 - creating public in Liferay Portal 6.x 21–23
- portal pages and portlets 178–193
 - Announcements portlet 182–183
 - built-in Document Library and Book Catalog portlets 188–191
 - Home portal page 178–181
 - RSS feed portlet 184–185
 - setting permissions for 185–187, 191–193
- server features and limitations 162–163
 - directly using portal databases 163
 - reusing built-in services 162–163
- servers 16–18
- server-specific APIs 448–450
- PortalService implementations 164
- PortalService interface 164
- portal-service.jar JAR file 448
- Portlet 2.0, support for Ajax in 450–455
 - sending resource request using resource URL 453–455
 - serving resources 450–453, 455
- portlet API 69–70, 377
- portlet applications 11, 240, 251
 - container-runtime options 130–131
 - sharing session data across 119
- portlet bridges 14, 95, 510–532
 - description of 511–513
 - iFrame portlets 513–515
 - JSF portlets 515–525
 - Wicket portlets 525–532
- portlet class 148
 - extends GenericPortlet class 64–67
 - implements portlet interface 64
 - obtaining form field values in 227–228
 - GlassFish server 228
 - Jetspeed portal 228
 - Liferay Portal portal 227–228
 - obtaining preferences in 397–400
- portlet configuration, retrieving using Portlet-Config object 120–121
- portlet container 13, 15
- portlet content 254, 440
- portlet deployment descriptors 392–395, 523–525
- <portlet> element 37–38, 72, 76, 410–411
- portlet environment, discovering using Portlet-Context object 119–120
- portlet events 409, 425–432
 - advantages and disadvantages of 435–436
 - receiving 432–435
 - defining events that portlet can process 432–433
 - implementing EventPortlet interface 433–435
- sending 427–432
 - configuring 430–431
 - defining events in portlet deployment descriptor 431–432
 - setting events for publishing 428–430
 - support in Spring Portlet MVC interface 436
- portlet filters 494–509
 - converting text to hyperlinks with 505–508
 - filter chaining 508–509
 - interface methods 496–497
 - lifecycle 497–499
 - setting portlet title with 500–503
 - types of 495–496
 - using with Book Catalog portlet 499–500
 - validating requests with 503–505
- portlet handles 537
- portlet icon 184

- Portlet interface 57–60, 92, 433
 - destroy method 59
 - init method 58
 - portlet class implements 64
 - processAction method 59
 - render method 58–59
- Portlet Management interface 537
- <portlet mode name> 65
- portlet modes 61–62
- portlet namespace 487
- portlet objects, making available with
 - <defineObjects> tag 210–212
- portlet request objects 92–100
 - and lifecycle interfaces 92–94
 - handling portal server-specific requests using PortalContext object 99–100
- PortletRequest constants 94–97
 - ACTION_SCOPE_ID 95
 - LIFECYCLE_PHASE 94–95
 - RENDER_PART, RENDER_HEADER, and RENDER_MARKUP 95–96
 - USER_INFO 96–97
- PortletRequest methods 97–98
- portlet request-processing lifecycle 395
- portlet requests, retrieving preferences from 382–383
- portlet response objects 100–112
 - adding custom JavaScript and CSS files to portal page 108–112
 - MARKUP_HEAD_ELEMENT support 108–111
 - portal server-specific configuration 111–112
 - choosing appropriate phase for action 107–108
 - encoding URLs with encodeURL method 102–103
 - generating content using RenderResponse object 103–104
 - responding to actions using ActionResponse object 104–107
- portlet sessions, storing user-specific data in 112–119, 409
 - and content caching 118
 - obtaining session attributes using PortletSessionUtil class 114–116
 - portlet session scopes 116–118
 - PortletSession object 112–114
 - sharing session data across portlet applications 119
 - similarities between HttpSession and PortletSession objects 118–119
- portlet tag library 467
- portlet titles, setting with portlet filters 500–503
- portlet URLs 76–78
- <portlet_name> element 244
- PORTLET_SCOPE session attributes 112–113, 115–117, 128
- portlet.title 104
- portlet.title.addBookForm property 502
- portlet.title.default property 502
- portlet.title.help property 502
- portlet.title.preferences property 502
- portlet.xml file 35, 409, 418–419, 456, 502, 507–508
 - defining preferences in 403–404
 - defining support for custom portlet modes in 144–145
- <portlet-app> element 37, 72, 161, 411, 431–432
- PortletApplicationContextUtils class 351–352
- portletAttr attribute 116, 128
- <portlet-class> element 37, 73
- portlet-column-first CSS class 205
- PortletConfig interface 120
- PortletConfig method 121
- PortletConfig object 75–76, 119–121, 432
- portletConfig object 213
- portletConfig variable 211
- PortletContext interface 452, 491–493, 497
- PortletContext mirrors 120
- PortletContext object 15, 69, 73, 112, 119–120
- PortletException 254
- PortletFileUpload class 157, 332
- PortletFilter interface 496–499
- <portlet-info> element 244
- Portlet-level permission 193
- portlet-managed portlet modes 148–150
- portletMode attribute 214
- PortletMode class 64
- <portlet-mode> element 37, 67, 144, 149, 380
- <portletMode> portlet mode 524
- PortletModeException 146
- PortletModeHandlerMapping bean 241, 245–246, 248, 257, 264
- PortletModeHandlerMapping
 - implementation 263–264
- portletModeMap property 257, 263
- PortletModeNameViewController controller 257–258
- PortletModeParameterHandlerMapping 267–271, 275
- PortletMultipartResolver bean, uploading files using 332–333
- <portlet-name> element 38, 73, 249–250, 503
- <portletName>-portlet.xml 341
- <portlet-preferences> element 393, 403
- PortletPreferences object 453
 - saving preferences with 384–388
 - atomicity 386
 - concurrency issues 386
 - resetting preferences 386–388

- PortletPreferences object (*continued*)
 - validating preferences before setting in 389–390
- PortletPreferences objects 525
- portletPreferences scripting variable 399–400
- portletPreferences variable 211
- portletPreferencesValues variable 211, 400, 402
- PortletRequest constants 94–97
 - ACTION_SCOPE_ID 95
 - LIFECYCLE_PHASE 94–95
 - RENDER_PART, RENDER_HEADER, and RENDER_MARKUP 95–96
 - USER_INFO 96–97
- PortletRequest interface 94, 96–97, 154, 295, 449
- PortletRequest methods 97–98, 192, 423
- PortletRequest object 95
- PortletRequest.USER_INFO constant 97
- PortletRequestDispatcher 73–74, 243, 248, 380
- PortletResponse interface 101–102, 418
- PortletResponse object 95, 456
- portlets 3–47, 178–193
 - adding to pages, in Liferay Portal 6.x 23–24
 - Ajax 440–447
 - portlets that display date and time 442
 - requests 442–445
 - retrieving servlet response to update portlet content 445–447
 - announcements 182–183
 - behavior of 135–136
 - built-in
 - changing locales using 143
 - Document Library and Book Catalog 188–191
 - impact of using 190–191
 - class, Hello World 36–37
 - combining databases and 335–336
 - components of, handling requests using 444–445
 - container-runtime options 130–131
 - containers 51–52
 - content of, retrieving servlet response to update 445–447
 - creating portal pages and adding to 568–570
 - custom
 - importing into GateIn Portal server 571
 - programmatically securing 570–571
 - defined 4
 - deployment descriptors 243–244
 - description of 10–11
 - development 68–80
 - container-runtime options 78–80
 - creating JSPs to display portlet content 70–71
 - deployment descriptor 72–73
 - dispatching portlet requests 73–74
 - initialization parameters 76
 - internationalization and localization 75
 - Portlet API objects 69–70
 - URLs 76–78
 - development environment 26–35
 - configuring Eclipse IDE 26–33
 - project structure 34–35
 - downloading binary content using 455–458
 - exposing Book Catalog web application 523–525
 - generating content based on portlet mode 62–68
 - defining support for portlet modes 67–68
 - writing content generation methods 64–67
 - Hello World 35–47, 241–246
 - adding to portal page 43–46
 - controller class 242–243
 - JSP page 243
 - Liferay Portal 6.x 38–39
 - portlet class 36–37
 - portlet deployment descriptor 243–244
 - portlet settings 37
 - resource bundle 37–38
 - web application context 244–245
 - web application deployment descriptor 245–246
 - with Ant tool 39–41
 - with Maven tool 42–43
 - implementing to support localizing 142
 - infrastructure of 15–18
 - portal server 16–18
 - portlet container 15–16
 - lifecycle phases of 295
 - lifecycle, vs. servlet lifecycle 51–54
 - Liferay Portal 6.x 18–26
 - adding portlets to pages 23–24
 - creating public portal page in 21–23
 - installing 18–20
 - registering users with 20
 - roles and permissions in 24–26
 - modes 143–151
 - and markup types 150–151
 - custom 144–147
 - defining support for 67–68
 - portal-managed and portlet-managed 148–150
 - setting in URL 147
 - setting programmatically 145–146
 - portability of 16
 - portals
 - benefits of 6–10
 - description of 4–6
 - portlet class
 - extends GenericPortlet class 64–67
 - implements portlet interface 64

- portlets (*continued*)
 - real-time 480–488
 - DWR library integration options 488
 - using Comet feature 481–482
 - using polling 481
 - reasons to use 12–15
 - inter-portlet vs. inter-servlet communication 12–13
 - mashups 15
 - portlets vs. widgets 13–14
 - remote
 - definition of 535–538
 - locally registering 547–550
 - rendering even when minimized 255
 - RSS feed 184–185
 - setting permissions for 185–187
 - book catalog 191–192
 - discussion forums 192–193
 - settings, Hello World 37
 - tag library 207–229
 - <actionURL> tag 213–221
 - <defineObjects> tag 210–213
 - <namespace> tag 223–228
 - <param> tag 221–223
 - <property> tag 228–229
 - <renderURL> tag 220
 - <resourceURL> tag 220–221
 - using to build Book Catalog portlet 208–209
 - that display date and time 442
 - undeploying applications 46–47
 - URLs 54–57
 - types of 56–57
 - vs. servlet URLs 54–56
 - window states, and markup types 153–154
 - PortletSession attributes 118–119, 126–127
 - PortletSession interface 409–417, 447–449, 452
 - advantages and disadvantages of 416–417
 - defining multiple portlets 410–411
 - example 415–416
 - storing and retrieving information from 411–415
 - caching and inter-portlet communication 413–415
 - choosing which information to communicate 411–413
 - PortletSession object 98, 113
 - and methods provided by 112–113
 - HttpSession object, similarities between 118–119
 - usage in Book Catalog portlet 113–114
 - PortletSession property, specifying whether required 254
 - portletSession variable 211, 213
 - PortletSession-based communication 409
 - portletSessionScope variable 211, 213
 - PortletSessionUtil class, obtaining session attributes using 114–116
 - PortletTitleFilter filter 500–503
 - PortletURL method 223
 - PortletURL object 147
 - PortletURLGenerationListener 156, 159–161, 223, 229
 - PortletWrappingController controller 259–260
 - portlet-xhtml directory 517
 - POST method, HTTP 499, 503, 505
 - POST request, HTTP 444
 - postRenderHandle method 274
 - PowerUser role 195
 - pref1 preference attribute 387, 398–399
 - pref2 preference attribute 387
 - prefBook key 389
 - prefBookISBN preference 391, 394–395
 - prefCategory request parameter 382
 - <preference> element 403–404
 - <preference-name> element 397
 - preferences 382–384
 - behavior 402–403
 - content 400–402
 - defining in portlet.xml file 403–404
 - retrieving
 - from portlet request 382–383
 - obtaining preferences in portlet class and JSP pages 397–400
 - saving
 - to persistent store 383–384
 - with PortletPreferences object 384–388
 - validating 388–397
 - before setting in PortletPreferences object 389–390
 - handling ValidatorException exception 395–397
 - with PreferencesValidator interface 390–392
- Preferences dialog box 31
- preferences.jsp 70
- PreferencesValidator classes 386, 388, 393–395
- PreferencesValidator interface
 - configuring in portlet deployment descriptor 392–395
 - validating preferences with 390–392
- <preferences-validator> subelement 393
- preference-validation logic 390
- Preferred Book ISBN Number preference 400
- Preferred Category preference 400
- prefix property 245
- preHandle method 273
- preHandleAction method 275
- PreparedStatement objects 345, 348
- PreparedStatementSetter interface 348–349
- preRenderHandle method 274
- preventCache property 466

PrintWriter object 36, 73, 101, 505, 507
 private constructor 364
 Private Pages option 22
 private render parameters 216–220, 417
 <private-session-attributes> element 448
 proceed method 357
 ProceedingJoinPoint object 357
 process events function 108
 processAction method 58–60, 77–79
 processAddedBookEvent method 434
 processEvent method 398, 404, 433–434, 436, 495
 Producers hyperlink 541
 Producers tab 542
 producers, WSRP 550–554
 Profile object 472
 ProfileBean object 472–473
 ProfileBean.js file 473
 profiles, server 26
 programmatic security 154–156, 192
 project element 39
 Properties tab 560
 properties, adding to URLs using <property>
 tag 228–229
 property element 239
 <property> tag 208, 228–229
 PropertyColumn<Book> class 529
 PropertyEditor 307
 PropertyEditorRegistrars 309–310
 PropertyPlaceholderConfigurer bean 340–341
 proxy JavaScript object 473
 public page 21
 Public Pages option 22
 Public Pages tab 179
 public portal pages 21–23, 178
 public render parameters 409, 417–425
 advantages and disadvantages of 424–425
 example 425
 methods related to 423–424
 setting and retrieving 418–423
 public static void method 327
 public-render-parameter element 428
 <public-render-parameter> elements 420, 423
 publishing, setting events for 428–430
 PUT request, HTTP 444

Q

<qname> element 420, 423, 428, 432
 QNames 432
 query method 348

R

README.TXT file 344, 558
 <read-only> element 404

read-only option 378
 readyState property, XMLHttpRequest
 object 445–447
 real-time portlets 480–488
 DWR library integration options 488
 using Comet feature
 scalability problem and solution 482
 support for in DWR library 482–487
 using polling 481
 receiver portlet 411
 Recent Book page 416
 recentBookIsbn attribute 414, 420–422
 recentBookIsbn name 423
 recentBookIsbn render parameter 420–423
 recentBookIsbn session attribute 414
 Recently Added Book portlets 410, 416, 418–419,
 428, 489–491
 RecentlyAddedBookPortlet class 414–415, 492
 recentlyAddedBookPortletNamespace
 attribute 492–493
 recentPortletHome.jsp page 414
 redirect user to a different page function 107
 ref subelement 257
 reference data 301
 Refresh hyperlink 441, 443, 447, 453–454
 Refresh Search Results 137
 region attribute 464
 registerCustomEditor method 308–309
 registered users 168–169
 defining default groups and memberships
 for 572–573
 to book catalogs 192
 registering
 locally, remote portlets 547–550
 users with Liferay Portal 6.x 20
 WSRP consumers with WSRP producers
 in-band 551–553
 leasing 553–554
 out-of-band 551–553
 registerUser method 67, 70–71, 79, 83, 85
 registerUserActionUrl 82, 84
 registration handle 551
 Registration interface 537
 Registration Property Descriptions section 552
 registration requirements 538
 Registration Type combo box 553
 registrationForm.jsp 70
 Regular Organization 171
 Regular Roles 177
 relational database tables, mapping classes
 to 363–365
 ReloadableResourceBundleMessageSource
 implementation 330
 remote files, CSS and JavaScript URLs for
 549–550

- Remote Portlet combo box 547
- remote portlets
 - definition of 535–538
 - locally registering 547–550
- <remoteable> element 543
- Remove link, Book Catalog portlet 460
- removeAttribute method 119
- removeBook methods 347, 349–350, 358, 360, 370
- RemoveBookController 301
- removePublicRenderParameter method 423–424
- render methods 53–54, 58–61, 81–83, 99, 294–295, 451–452
 - BookCatalogPortlet class 139
 - DateTimePortlet class 448
 - identifying using @RequestMapping annotation 293–295
 - identifying with @RenderMapping annotation 295–297
 - and method signatures 297
 - options for 296–297
 - writing for EDIT portlet mode 380–381
- render parameters
 - adding private to <actionURL> tag 216–220
 - public 417–425
 - advantages and disadvantages of 424–425
 - example 425
 - methods related to 423–424
 - setting and retrieving 418–423
- Render portlet 52
- render URLs 77
- RENDER_HEADER constant 95–96
- RENDER_MARKUP constant 95–96
- RENDER_PART constant 95–96
- RenderFilter interface 495, 497–498, 500–501, 505
- renderForm method 67, 71, 78–79, 83
- renderHeaders container-runtime option 122, 130
- renderHelp method 82
- rendering, portlets even when minimized 255
- renderPrefs method 67, 82
- RenderRequest argument 294
- RenderRequest object 64, 123, 125, 127, 300, 497
- renderRequest variable 210, 213
- RenderRequestWrapper object 498
- RenderResponse argument 294
- RenderResponse interface 101
- RenderResponse object 69, 76–77, 101, 207, 210, 453–454, 504
 - generating content using 103–104
 - getWriter method 506
 - setTitle method 500–501
- renderResponse variable 210, 213
- RenderResponseWrapper 158, 498, 505–506
- <renderURL> tag 208, 213, 216, 220, 222, 228
- renderURL tag 467
- renderWhenMinimized property 255
- replacements list 508
- reqParamName initialization parameter 501, 503
- request attribute 454
- request handling 277–278
- request objects, portlet 92–100
 - and lifecycle interfaces 92–94
 - handling portal server-specific requests using PortalContext object 99–100
 - PortletRequest constants 94–97
 - PortletRequest methods 97–98
- request parameters 285, 503
 - accessing namespaces
 - by adding values 228
 - with or without values 228
 - without values 227–228
 - passing using @RequestParam annotation 299–301
 - options for 300–301
- request processing
 - in Spring Portlet MVC framework 246–248
 - synchronizing 255–256
 - using handler interceptors 273–276
- request types 52
- request variable 207
- request-handling logic 163
- request-handling method 294
- RequestMapping annotation 290
- request-processing methods 398
- RequestProcessor class 246
- requests
 - Ajax
 - handling requests using portlet or servlet components 444–445
 - sending requests using XMLHttpRequest object 442–444
 - multiple phases of processing 52–54
 - portlet
 - dispatching using PortletRequestDispatcher object 73–74
 - retrieving preferences from 382–383
 - resource, sending using resource URL 453–455
 - securing 447–450
 - date/time portlet security 447–448
 - portal server-specific APIs 448–450
 - validating with portlet filters 503–505
- requestScope JSP implicit variable 213
- require function 465
- required element 300
- requireSession property 254
- Reset hyperlink 49
- Reset links 378
- reset method 385–388
- resetPreference method 387
- resetRenderUrl 82

resolveException methods 276
 resource bundles 37–38, 75
 <Resource> element 343
 resource identifier 451
 resource requests, sending using resource
 URL 453–455
 resource URLs (Uniform Resource Locators)
 and caching 458–459
 sending resource requests using 453–455
 RESOURCE_PHASE 295
 ResourceAwareController 260
 <resource-bundle> element 37, 73, 410
 ResourceBundleMessageSource 330
 <resource-bundle-name> element 142
 ResourceBundleViewResolver 247
 ResourceFilter interface 495
 <ResourceLink> elements 342–344
 ResourceRequest interface 93
 ResourceRequest object 451–452, 454
 resourceRequest variable 211
 ResourceResponse interface 101
 ResourceResponse object 451, 453–454, 457
 resourceResponse variable 211
 resources
 caching 221
 identifying 221
 serving
 with ResourceServingPortlet interface
 450–453
 with Spring Portlet MVC 455
 resources directory 517–518
 ResourceServingPortlet interface 92, 220–221,
 437, 450, 453, 455
 ResourceURL method 223
 ResourceURL object 454, 458
 <resourceURL> tag 208, 214, 220–222, 453
 resourceURL tag 458, 467
 response data, from servlet 445–447
 response objects, portlet 100–112
 adding custom JavaScript language and CSS files
 to portal page 108–112
 choosing appropriate phase for action 107–108
 encoding URLs with encodeURL method
 102–103
 generating content using RenderResponse
 object 103–104
 responding to actions using ActionResponse
 object 104–107
 response.setRenderParameter 107
 responseData attribute 479
 responseMessage attribute 479
 responseText method, XMLHttpRequest
 object 446
 responseXML method, XMLHttpRequest
 object 446

ResultSet object 339, 345, 348, 361
 rich user interfaces 461–480
 for Book Catalog portlet 459–461
 simplified Ajax with Dojo Toolkit 462–468
 adding widgets to JSP page 463–465
 Ajax support 465–468
 including necessary JavaScript and CSS in por-
 tal page 463
 simplified Ajax with DWR library 471, 480
 configuring DWR library 473, 476
 including necessary JavaScript files 476–477
 simplified Ajax with jQuery library 468, 471
 role information, in ROLE_ table 579
 _ROLE table 449
 ROLE_ table, role information in 579–580
 <role-link> element 155
 RoleLocalServiceUtil class 448–449
 <role-mapper> element 155
 <role-name> element 155
 roles
 creating for Book Portal portal users 173–174
 custom, associating with organization 174–175
 in Liferay Portal 6.x 24–26
 mapping to Book Portal portal
 organization 174–175
 mapping users to 177–178
 root web application contexts 241, 251–252
 RowMapper callback interface 348
 RowMapper interface 348
 RSS built-in portlet 181
 RSS feed portlet 184–185

S

Save button 543
 Save link, Book Catalog portlet 461
 save method 367, 369
 savePreferences method 382–384, 389, 393, 395
 sayHello method 37
 scalability, problem and solution 482
 schema-based approach 354
 scopes, portlet session 116–118
 script DOM element 110
 <script> element 108–110, 202
 script element 110
 script session listener 483
 script session manager 483
 <script> tag 225, 550
 ScriptBuffer object 487
 scripts, Ant tool 39–40
 ScriptSession object 483–487
 ScriptSessionManager object 483–485
 scriptSessions variable 487
 search portlet mode 150

- search type preference, personalizing behavior
 - based on 402–403
- searchBooks method 114, 402–403
- SearchCriteria object 472
- searchNames list 508
- searchProfile method, ProfileBean object 472–473
- searchType preference 403
- secure attribute 216
- securing, custom portlets 570–571
- security
 - of <actionURL> tag 216
 - of requests 447–450
 - date/time portlet security 447–448
 - portal server-specific APIs 448–450
 - programmatic 154–156
- securityInterceptor 273
- Select Permission button 561
- send events to other portlets function 108
- send method 443, 446
- sender portlet 411
- sendRedirect(String location) method 105
- Serializable interface 429
- Server category 540
- server.xml files 342, 481–482
- serveResource method 454, 466–469, 489–491
- servers
 - configuring settings for Liferay Portal 193–196
 - setting authentication mechanism 194–195
 - specifying default user associations 195–196
 - for Eclipse IDE 28–30
 - portal server-specific configuration 111–112
 - portal, features and limitations 162–163
 - profiles, creating new 26
 - Tomcat, specifying installation directory of 27–28
- server-specific APIs, portals 448–450
- Service Description (interface) 537
- service implementation classes 349–351
- service interface classes 349–351
- service layer 163
- service method 52, 54
- service-oriented architecture. *See* SOA
- services and DAOs 518
- servicing, resources
 - with ResourceServingPortlet interface 450–453
 - with Spring Portlet MVC 455
- Servlet 3.0, asynchronous request processing 482
- Servlet interface 57
- ServletContext attributes 120, 413, 422, 425–426
- ServletContext listeners 120
- ServletContext mirror 251
- ServletContext object 114, 493
- servletDefaultSessionScope container-runtime option 122, 127–129
- <servlet-mapping> element 55
- servlets
 - components of 444–445
 - containers 51–52
 - inter-portlet vs. inter-servlet communication 12–13
 - lifecycle, portlet lifecycle vs. 51–54
 - retrieving response data to update portlet content 445–447
 - URLs, portlet URLs vs. 54–56
- session attributes, obtaining using PortletSessionUtil class 114–116
- session data, sharing across portlet applications 119
- session JSP variable 213
- Session Manager window 575
- Session object 370–371
- session variable 207
- session.shared.attributes property 417, 448
- SessionFactory interface 367–371
- sessionScope JSP implicit object 210
- SessionStatus object 306, 327
- set expiration timestamp function 107
- set render parameters function 107
- set window state and portlet mode function 107
- setActiveReverseAjax function 485
- setAttribute method 110, 113, 115, 119, 218
- <setBundle> tag 84
- setBundle tag 142
- setCacheability method, ResourceURL object 458
- setComplete method 306
- setCurrentDateTime JavaScript function 443, 446
- setDisallowedFields method 308
- setETag method 139–140
- setEvent methods 428–429
- setExpirationTime method 135
- setMessageSource method 331
- setNextPossiblePortletModes method 104
- setNotifyServerOnPageUnload function 485
- setOrderService method 239, 251
- setParameter method 217
- setParameter(...) method 267
- setPortletMode method 145, 147, 150
- setProperty method 366, 368
- setRenderParameter method 216, 275, 417, 419, 435
- setResourceID method, ResourceURL object 454
- setScope method 135
- setSecure method, PortletURL class 160
- Settings option 194
- Settings page 195
- setTitle method 104, 500–501
- setUp method 327
- setUseCachedContent method 139–140
- setValue method 384–385, 387–388, 395

- setValues method 383–385, 387, 395
- showAddBookForm method 286, 296, 303
- showBooks method 106–107, 401–402
- showBooksController handler 266, 279–280
- showBookUpdateMsg function 470–471, 485, 487
- showDateTime method 445
- showEditBookForm method 305
- showHomePage render method 454
- showPrefs method 380–381
- showPrint render method 147
- showPrintableCatalog method 147
- showRecentBook method 414
- showSearchResult method 150
- showSomething method 292, 294, 296–297
- Sign In built-in portlet 181
- SimpleMappingExceptionResolver bean 278
- SimpleMappingExceptionResolver maps 278
- simplified Ajax
 - with Dojo Toolkit 462–468
 - adding widgets to JSP page 463–465
 - Ajax support 465–468
 - including necessary JavaScript language 463
 - with DWR library 471–480
 - configuring 473–476
 - including necessary JavaScript files 476–477
 - with jQuery library 468–471
- SOA (service-oriented architecture) 9, 12
- SomeController class 291, 299, 331
- SomeController handler 285, 291–293, 297–298
- SomeObject model attribute 314
- Soria theme 463
- soria.css file 463
- SortableDataProvider 529
- source code, compliance setting for Eclipse IDE 33
- Spring AOP framework 354
- Spring form tag library 282
- Spring framework
 - accessing Spring beans from non-Spring objects 351–352
 - AOP support in 352–358
 - transaction management support in 358–361
 - annotation-driven 360–361
 - defining transaction manager 358–359
 - specifying transactional methods 359
 - using with Hibernate ORM framework 367–372
 - AnnotationSessionFactoryBean bean 368–370
 - HQL 370–372
- Spring IoC container 240
- Spring JDBC classes 338
- Spring JDBC module 338
- Spring Portlet MVC 543
 - servicing resources with 455
 - support for portlet events in 436
- Spring Portlet MVC annotations 282
- Spring Portlet MVC concepts 284
- Spring Portlet MVC controllers 285, 426
- Spring Portlet MVC Date/Time Portlet 549
- Spring Portlet MVC framework 233–234, 260–333, 339
 - annotations 285–310
 - classpath scanning feature and autoregistration 286–288
 - configuring beans 310–312
 - dependency injection 288–290
 - developing Book Catalog portlet using 283–284
 - handling exceptions with
 - @ExceptionHandler 307
 - identifying action methods with
 - @ActionMapping 297–299
 - identifying controllers with @Controller 285–286
 - identifying model attributes using
 - @ModelAttribute 301–304
 - identifying render methods with
 - @RenderMapping 295–297
 - mapping requests to handlers and methods using @RequestMapping 290–295
 - overview of 282–283
 - passing request parameters using
 - @RequestParam 299–301
 - WebDataBinder object 307–310
 - application context XML file 340
 - built-in controllers 252–260
 - AbstractController class 254–256
 - Controller interface 253
 - ParameterizableViewController controller 256–257
 - PortletModeNameViewController controller 257–258
 - PortletWrappingController controller 259–260
 - built-in handler mappings 260–272
 - AbstractHandlerMapping class 270–272
 - developing Book Catalog portlet 262
 - ParameterHandlerMapping implementation 264–267
 - PortletModeHandlerMapping implementation 263–264
 - PortletModeParameterHandlerMapping implementation 267–270
 - configuring JSR 303 318–321
 - DI and application context 235–241
 - e-commerce example 235–237
 - web application contexts 240–241
 - DispatcherPortlet class 248–249
 - external JavaScript and CSS files 331–332
 - form tag library 323–324

- Spring Portlet MVC framework (*continued*)
 - handler interceptors 272–276
 - implementing and configuring 272–273
 - processing requests using 273–276
 - handling exceptions 276–280
 - during request handling 277–278
 - mapping to view based on exception class 278–280
 - Hello World portlet 241–246
 - controller class 242–243
 - JSP page 243
 - portlet deployment descriptor 243–244
 - web application context 244–245
 - web application deployment descriptor 245–246
 - localizing content 330–331
 - reasons for using 234–235
 - request processing in 246–248
 - TestContext framework of, unit testing with 327–330
 - unit testing 324–330
 - with Ant buildfile 325–326
 - with mock objects 326–327
 - with Spring TestContext framework 327–330
 - uploading files using PortletMultipartResolver bean 332–333
 - validating forms 313–323
 - validating beans with JSR 303 316–323
 - with Spring Validation API 313–316
 - web application context 249–252
- Spring Portlet MVC-based portlets 241
- Spring TestContext framework 282
- Spring Web MVC framework 335
- Spring-based applications 340
- spring-context schema 287
- SpringJUnit4ClassRunner 328
- Spring-specific JAR files 241
- SQL statements 338
- SQLException 349
- src directory 34
- startBookUpdateThread method 474
 - BookCatalogAjaxBean 485
 - BookCatalogAjaxBean object 484–485
- StateAwareResponse interface 101, 423, 428
- StateException 278, 280
- Statement object 339, 345
- states, window 151–154
 - custom 152–153
 - portlet 153–154
 - specifying 214
- storage
 - of information from PortletSession interface 411–415
 - caching and inter-portlet communication 413–415
 - choosing which information to communicate 411–413
 - of user-specific data, in portlet session 112–119
 - store method 384–386, 388–389
 - strategy element 364
 - streaming-based portal 95
 - string concatenation, adding parameters to portlet URLs with 223
 - String element 288, 294
 - String getValue method 385
 - String Java type 367
 - String object 243, 430
 - String parameter 308
 - String type 480
 - String values 239–240, 294, 435
 - Struts Action 512
 - style attribute 471
 - submit forms function 107
 - success parameter, ajax function 469
 - success.jsp 70
 - support, defining for portlet modes 67–68
 - <supported-processing-event> element 433
 - <supported-public-render-parameter> element 420–421
 - <supported-publishing-event> element 432–433
 - <supports> element 150, 380
 - synchronizeOnSession property 255
 - synchronizing request processing 255–256

T

- TabContainer layout widget 462–464
- tables, Liferay Portal platform 579–581
- tag library 207–229
 - <actionURL> tag 213–221
 - adding private render parameters to 216–220
 - escaping XML in generated 220
 - securing 216
 - specifying action name 220
 - specifying portlet mode 214
 - specifying window state 214
 - storing in variable 214–215
 - <defineObjects> tag
 - making portlet objects available with 210–212
 - using in Book Catalog portlet 212–213
 - <namespace> tag 223–228
 - <param> tag 221–223
 - <property> tag, adding properties to URLs using 228–229
 - <renderURL> tag 220
 - <resourceURL> tag 220–221
 - using to build Book Catalog portlet 208–209
- taglib directive 84, 212
- tags, form tag library 323–324
- <td> tag 205

- templates, page layout 203–206
- test directory 34
- testBook method 329
- TestContext framework, of Spring Portlet MVC framework 327–330
- testShowAddBookForm 327
- text value 466
- text, converting to hyperlinks 505–508
- text/vnd.wap.xhtml+xml markup 151, 154
- themes folder 197
- themes, for portals 198–203
 - adding JavaScript files as part of 202
 - changing background color of portal page 200–201
 - changing default breadcrumbs image 201
 - deploying and setting 202–203
- threads 481
- Timeouts tab 30
- TITLE column 337
- title.properties file 501–502
- titles, portlet 500–503
- TOC (table of contents) 335, 337, 358, 455
- TOC files 218, 456–457
- TOC_FILE column 338
- TOC_ID column 337
- TOC_TBL tables 336–338, 349, 351, 358
- toc_tbl.sql 338
- TocDao 351
- ToCFile object 477, 480
- tocFile property 480
- Tomcat server, installation directories 27–28
- TOMCAT_HOME/conf/context.xml file 343
- TOMCAT_HOME/conf/server.xml file 343
- TOMCAT_HOME/lib directory 28, 343
- TOMCAT_HOME/lib/ext directory 448
- tomcat-6.x directory 19
- toString method 223
- transaction management, support in Spring framework 358–361
 - annotation-driven 360–361
 - defining transaction manager 358–359
 - specifying transactional methods 359
- transaction manager, defining 358–359
- transactional methods, specifying 359
- TransactionalTestExecutionListener 329
- transaction-manager attribute 360, 371
- try-catch blocks 349, 393
- type conversion 300
- type parameter, ajax function 469
- type-level @RequestMapping annotation 285, 292, 297
- types element 306

U

- UIs (user interfaces) 556–557
- undeploying, applications 46–47
- Uniform Resource Locators. *See* URLs
- unit testing 324–330
 - with Ant buildfile 325–326
 - with mock objects 326–327
 - with Spring TestContext framework 327–330
- unqualified name 364
- update method 348
- Update Service Description 547
- UpdateSender object 483–484, 486
- UpdateSender thread 484–487, 490
- upload files function 107
- uploadFolder initialization parameter 377, 456
- uploadFolder portlet 133, 158, 209, 456
- uploading files 156–158
 - Apache Commons FileUpload library 157–158
 - using PortletMultipartResolver bean 332–333
- uploadToc action method 107
- uploadTOC method 166
- uploadToc method 358, 360
- uploadTocActionUrl variable 218–219
- uploadToCFile function 479–480
- uploadTocFile method 474, 477, 480
- url element 201
- url parameter, ajax function 469
- url property 341, 443, 466
- URLs (Uniform Resource Locators)
 - action 77–78
 - adding properties to using <property> tag 228–229
 - encoding with encodeURL method 102–103
 - for remote CSS and JavaScript files 549–550
 - generation of listeners 159–161
 - portlet 54–57, 76–78
 - adding parameters to with string concatenation 223
 - types of 56–57
 - vs. servlet URLs 54–56
 - render 77
 - resource
 - and caching 458–459
 - sending resource requests using 453–455
 - setting portlet mode in 147
- Use Default Compliance Settings check box 33
- User ID 176
- user information, in USER_ table 579
- user interfaces. *See* UIs
- User Registration portlet 80–85, 549
 - JSP pages 84–85
 - requirements for 49–50
- User Repository 195
- User role 195, 447

- `_USER` table 449
- `USER_` table 579
- `USER_INFO` constant 96–97
- `user.login.id` user attribute 349
- `UserACL` class 570
- `<user-attribute>` elements 96–97
- user-defined metadata 361
- `<user-home>/m2/repository` 43
- `<user-id>` 197
- `<username>` element 378
- username property 341
- `userRegistration` portlet 130
- `UserRegistrationPortlet` class 81–83
- `userRoles` list 449
- users 563
 - defining default groups and memberships for registered 572–573
 - groups, associating users to 563–566
 - Guest, to book catalogs 191–192
 - of Book Portal portal, creating roles for 173–174
 - registering with Liferay Portal 6.x 20
 - specifying default associations 195–196
- `USERS_ROLES` table 449
- users, groups 566–567
- user-specific data, storing in portlet session 112–119
 - and content caching 118
 - obtaining session attributes, using `PortletSessionUtil` class 114–116
 - portlet session scopes 116–118
 - `PortletSession` object 112–114
 - sharing session data across portlet applications 119
 - similarities between `HttpSession` and `PortletSession` objects 118–119
- `util.js` file 476–477
- `UtilityFilter` class 503–505

V

- `validate` method 317, 320–321, 391, 393, 395
- `validateIsbn` method 522
- validating
 - forms 313–323
 - Spring Validation API 313–316
 - validating beans with JSR 303 316–323
 - preferences 388–397
 - before setting in `PortletPreferences` object 389–390
 - handling `ValidatorException` exception 395–397
 - with `PreferencesValidator` interface 390–392
 - requests, with portlet filters 503–505
- validation-based caching 137–141

- `ValidationMessages.properties` file 322
- `Validator` class 315
- `Validator` interface 318, 390
- `validator` property 318–319
- `ValidatorException` class 391–392, 396–397
- `ValidatorException` exception 395–397
- value attribute 223
- value element 286, 299–300, 455
- `<value>` elements 76, 128, 404
- `valueChangeListener` attribute 522
- values
 - form field, obtaining in portlet class 227–228
 - of namespaces
 - accessing namespaced request parameters 228
 - accessing request parameters with or without 228
- `<value-type>` element 432
- variables, storing `<actionURL>` tag in 214–215
- `VelocityViewResolver` 247
- `VIEW` mode 62, 68, 245, 290, 296, 380
- `View Page Properties` button 569
- `View Pages` button 180
- `View` permission 192
- `VIEW` portlet mode 442
- `ViewModeController` handler 263
- `ViewRendererServlet` configuration 245, 247–248, 276–277
- `ViewRendererServlet` servlet 245
- `viewRendererUrl` initialization parameter 246
- `ViewRenderServlet` servlet 247
- `ViewResolver` bean 247, 294, 307, 332
- `ViewResolver` interface 248, 250, 258, 277
- views
 - displaying when exception mappings not found 279–280
 - mapping to based on exception class 278–280
 - accessing exception as model attribute 280
 - exception mappings 278–280
 - handler mappings 279
 - handler-exception-resolver chain 280
- `void` action method 294
- `void` reset method 385
- `void` `setValue` method 385
- `void` `store()` method 385

W

- WAR containing 409
- WAR file 325, 555, 571
- `warningMsg` attribute 505
- `Weather` portlet 12
- web application contexts 240–241, 244–245, 249–252
 - root 251–252

- web application contexts (*continued*)
 - XML files
 - naming 249–250
 - scope of beans defined in 250–252
 - web applications, deployment descriptor 245–246
 - web pages, Wicket 528–529
 - Web Services for Remote Portlets. *See* WSRP
 - web.xml file 35, 475, 485
 - WebApplication class 528–529, 531
 - WebApplication object 529–532
 - WebApplicationContext object 249–252, 351–352
 - webapps 28
 - webapps directory 46
 - WebBindingInitializer object 310
 - webBindingInitializer property 310
 - WebClipping portlet 513
 - WebDataBinder initialization 285, 308–309, 314
 - WebDataBinder instances 308, 315, 324
 - WebDataBinder object
 - globally initializing instances of 309–310
 - initializing using @InitBinder annotation 307–308
 - WEB-INF directory 200
 - WEB-INF/jsp directory 34
 - WEB-INF/jsp/ folder 245
 - WEB-INF/jsp/edit.jsp JSP page 258
 - WEB-INF/jsp/parameterizedView.jsp page 257
 - WEB-INF/lib directory 34, 241, 523
 - WebLogicJtaTransactionManager 360
 - WebPage class 526, 528
 - WebSphere Portal 14, 16, 538
 - WebSphereUowTransactionManager 361
 - web-xhtml directory 517
 - wicket 528–529
 - id attribute 528
 - id value 529
 - Wicket portlets, Book Catalog web
 - application 525–532
 - deployment descriptor 530–531
 - developing using Wicket portlets 528–530
 - portlet deployment descriptor 531–532
 - project structure 526–527
 - requirements of 525–526
 - wicketFilterPath initialization parameter 531
 - WicketPortlet class 511, 531–532
 - widgets
 - adding to JSP page 463–465
 - portlets vs. 13–14
 - window states 151–154
 - custom 152–153
 - portlet, and markup types 153–154
 - specifying 214
 - windowState attribute 213
 - <window-state> element 152
 - WindowState object 255
 - work request parameter 299
 - wrapper classes 158–159
 - WSRP (Web Services for Remote Portlets)
 - 533–554
 - consumers 545–546, 550–554
 - Liferay Portal platform 538–539
 - remote portlets
 - definition of 535–538
 - locally registering 547–550
 - WSRP option 541, 545
 - WSRP portlet 540–541
 - WSRP specifications 535, 537
 - wsrp-portlet servlet 549
-
- X**
- xhr.open 443
 - xhrArgs object 466
 - xhrGet function 466
 - XHTML (eXtensible Hypertext Markup Language) files 520–523
 - xhtml directory 517
 - XML files
 - application context 340–342
 - naming 249–250
 - scope of beans defined in 250–252
 - XMLHttpRequest object 444–445, 447, 467, 472, 488
 - XMLHttpRequest phases 445–446
 - XmlViewResolver 247
 - XyzDAO class 240
-
- Z**
- ZIPCode public render parameter 424